Unified Notification Service

User Guide

Prepared by

Marios Vitos, European Dynamics

Søren Roug

July 2008

Version 1.6

***User manual***

Version management

|  |  |  |  |
| --- | --- | --- | --- |
| No | Date | Changes | Author |
| 001 | 11/08/2005 | First Draft | Marios Vitos |
| 002 | 17/08/2005 | First Revision | Marios Vitos |
| 003 | 22/08/2005 | Second Revision | Marios Vitos |
| 004 | 23/08/2005 | Final Version for Quality Review | Marios Vitos |
| 1.0 | 25/08/2005 | Final Version | Marios Vitos |
| 1.1 | 02/09/2005 | Users Administration addition (Not issued to EEA) | Marios Vitos |
| 1.2 | 19/10/2005 | Removal of Notification Profile types. Changes in filters management and vacation flag. | Marios Vitos |
| 1.3 | 13/03/2007 | Added notification template scripting | Søren Roug |
| 1.4 | 27/03/2007 | Described the available objects to the scriptlets | Søren Roug |
| 1.5 | 28/11/2007 | Removed the UI management section | Søren Roug |
| 1.6 | 23/07/2008 | Updated information on python templates | Søren Roug |

Contents

1. General principles [4](#_toc485)

2. Content feeds [5](#_toc506)

3. Authenticate user account [6](#_toc527)

4. UNS administration [8](#_toc538)

4.1. PUSH and PULL channel modes [8](#_toc539)

4.2. Manage PULL channels [8](#_toc543)

4.3. Manage PUSH channels [14](#_toc742)

4.4. Notification templates management [17](#_toc762)

4.5. Management and monitoring of notifications [24](#_toc987)

4.6. UNS configuration [28](#_toc1106)

5. UNS User FEATURES [43](#_toc1574)

5.1. General principles [43](#_toc1575)

5.2. Notification profile management [44](#_toc1581)

6. RPC user features [54](#_toc1851)

6.1. UNS XML-RPC [54](#_toc1857)

6.2. UNS RPC Users UI [56](#_toc1914)

# General principles

The Unified Notification Service (UNS) is part of the EEA Reportnet system. The purpose of UNS is to improve the cooperation and transparency as well as ease the retrieval of information from Reportnet. This is accomplished by UNS, a notification system, which is independent of specific data providers.

UNS defines some terms to describe the functionality of the application. The following terms are very common:

XML

The Extensible Markup Language (XML) was designed to provide a means for information to be served, received, and processed on the Web in the way that is now possible with HTML.

RSS

RSS is an acronym for Rich Site Summary (or Really Simple Syndication), an XML format for distributing news headlines on the Web, also known as syndication.

RDF

Resource Description Framework is a formal data model from the W3C using XML for the description of web resources using machine-readable metadata.

Reportnet Web Dashboards

The Web Dashboards Service is part of the EEA Reportnet architecture. It provides key stakeholders in international environmental reporting with dashboards presenting summary information harvested from Reportnet modules, as well as from other Agency systems based on predefined profiles. Each registered Reportnet user may define own dashboard profile allowing him/her viewing of desired Reportnet information at dashboard web page. URL of that page is Reportnet user’s WDB address. For an example user ‘pavlone’ has WDB address <http://dixie.eurodyn.com:8080/do/pavlone/dashboard>.

User roles

Roles are named groups of related privileges that Reportnet Unified Notification Service grant to users or other roles. Access to the Reportnet Unified Notification Service will be granted at four levels of users:

1. UNS Administrator: Is a registered user who is allowed to perform the UNS administrative operations. This user also may include privileges of the UNS User.
2. UNS User: Is a registered user who is able to manages own subscriptions in the UNS.
3. UNS RPC User: Is registered user who can access the UNS programmatically in order to create new PUSH channels or the push RDF formed data. This user is also able to see own PUSH channels by using the UNS user interface as well to create new one.
4. Anonymous User: Is an unregistered user who is only able to see the UNS welcome page.

Content channels

Content channels are used for harvesting event content data. There are two types of channels. PULL and PUSH channels. Pull channels are external web applications from which UNS is pulling data, whereas PUSH channels are used by external web applications to send events data to the UNS. Reportnet users will be able to subscribe themselves to these channels and receive notifications for any new events.

Subscription to a Content channel

A Reportnet user is able to subscribe to a channel, for receiving notifications that are allowed to his/her role.

# Content feeds

The UNS application can read and extract event data from documents where the data is stored in a structured manner. Such structured data is found in XML dialects such as RSS and Resource Description Framework (RDF).

The characteristic of both of these markup languages is that they are designed to store not only content but also additional information describing the content.

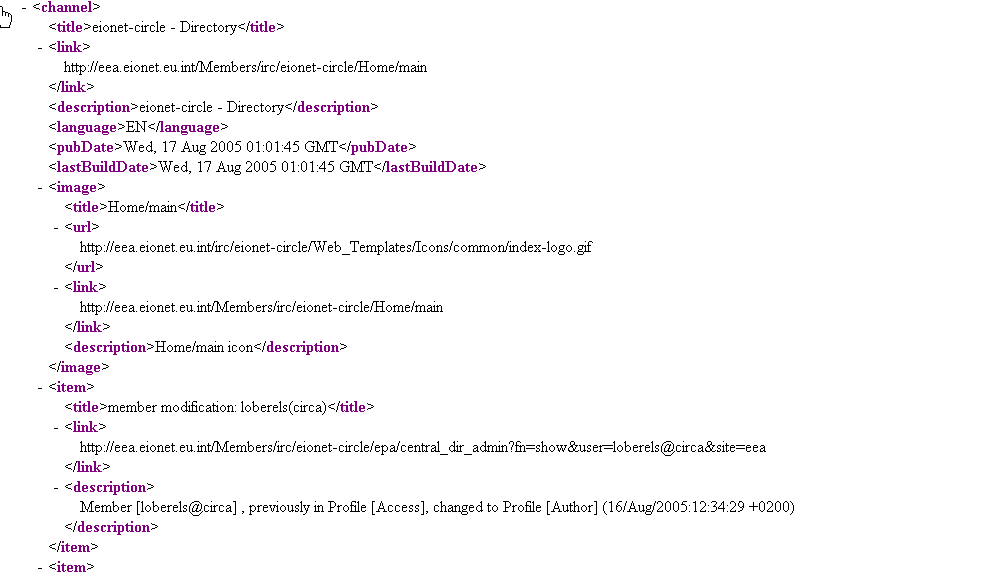
For example, in RSS, the content is stored in channels and the channel’s sub-elements: items.

A channel may contain any number of items which represent an event or a summary of the event and a link leading to the page with the full description of the event.

Each item contains some metadata elements that describe its content, such as[[1]](#footnote-2):

* title: the title of the item
* description: a summary/introduction of the item
* link: the URL where the full text is available
* author: the name/e-mail of the author of the item
* pubDate: the date when the item was published
* guid: a unique identifier for each item

A typical example of an RSS document is shown in figure 1:

Figure 1: Example of RSS document

These metadata elements are optional but at least a title or a description must be present in order for the content to be meaningful.

The same principles stand also for RDF type of documents. The only difference in the case of RDF is the syntax of the document itself[[2]](#footnote-3). Therefore RDF may provide other type of resources than simple RSS items.

UNS in turn, through its Harvester functionality, is able to browse through such structured documents and identify distinct resources. The system will store every recognized item and send a notification with all available metadata to UNS users (see section 4.2.1).

Additionally, it is possible to insert such structured documents into the UNS. This function is available only to event channels that operate under PUSH mode (see section 4.3)

UNS has a capability to handle all current versions of RSS and RDF in a way that is transparent to the user.

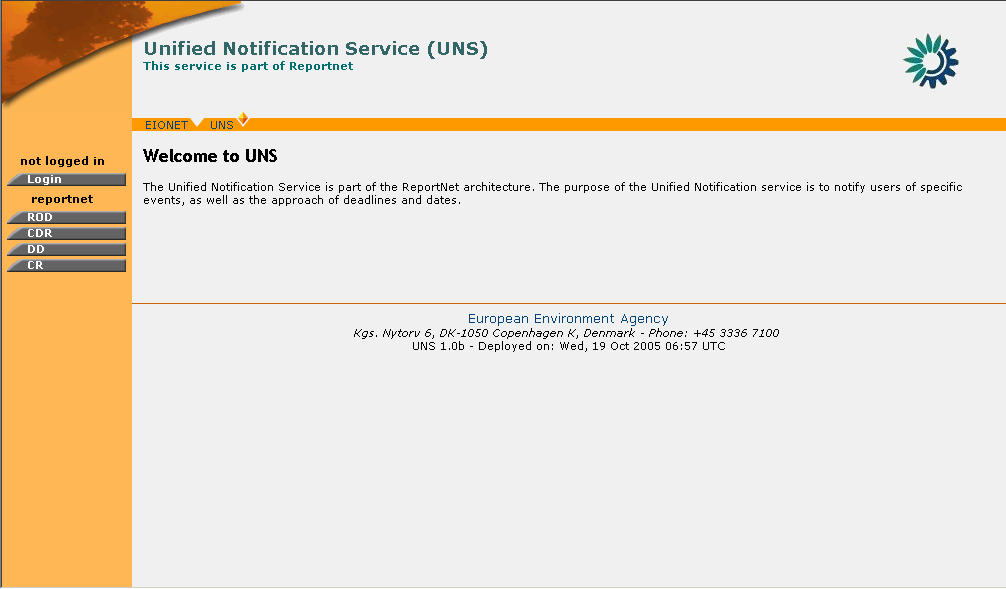
UNS allows the Reportnet users to retrieve their events through a RSS feed. Reportnet users can access this feed using any RSS/RDF reader software. This feed serves events collected during the last 5 days. Any RSS reader software could be configured to archive articles so the user will not loose old retrieved events.

# Authenticate user account

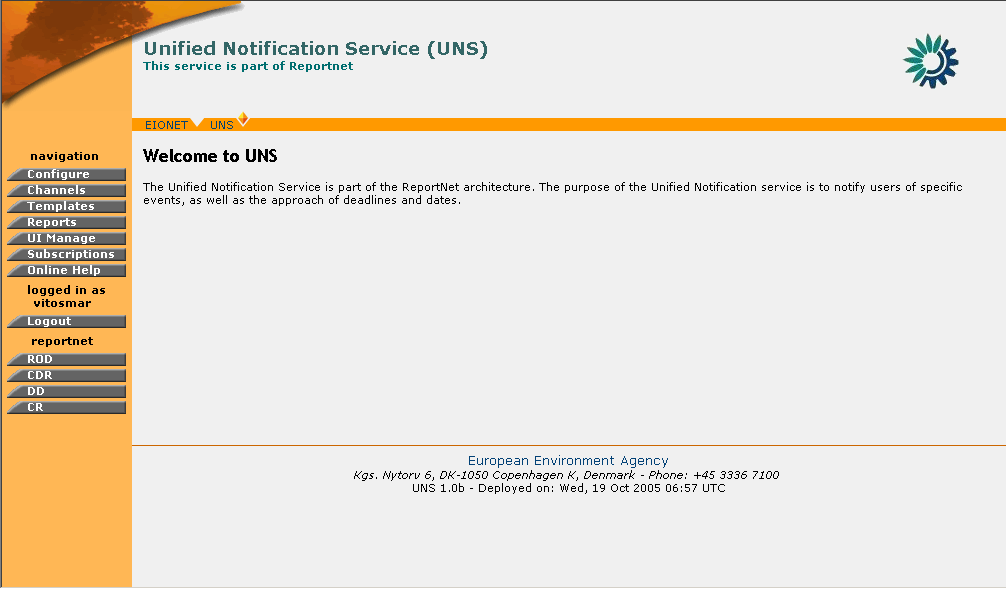
In UNS, users may log in the application in order to change their Notification Profile and subscription parameters. Administrators have additional privileges in terms of managing application parameters and defining new content feeds (See section 4). UNS XML-RPC users may login in the application in order to create new PUSH channels and to get list of PUSH channels they may access to.

Before performing any actions on UNS, you must authenticate yourself to the system as follows:

* The authentication page can be invoked by clicking on the “Login” icon at the top of the navigation bar
* Insert the username/password combination
* Click on the OK button

Figure 2: UNS Welcome Page

After a successful login, the navigation bar is refreshed with the operations corresponding to the profile of the user signed-in.

Figure 3: UNS Administrator welcome page

In order to logout, the user must select the “Logout” button on the navigation menu. By clicking on this button, the user will be disconnected from the application and the navigation menu will be reduced to the one shown in figure 2.

# UNS administration

## PUSH and PULL channel modes

The channels in UNS can operate in two modes; the PUSH mode and the PULL mode.

* Pull Mode: Channels operating under the PULL mode are feeds provided by services external to UNS. The application can access them and pull data from them (hence the name: PULL).
* Push Mode: Channels operating under the PUSH mode are created with the purpose of enabling users to insert events in them. UNS is designed to draw information about new events and send notifications to the Reportnet users.

## Manage PULL channels

### Create PULL content channel

A channel created in the “PULL” mode will be used to harvest information from external data feeds.

In order to create a channel you must be logged in as UNS Administrator.

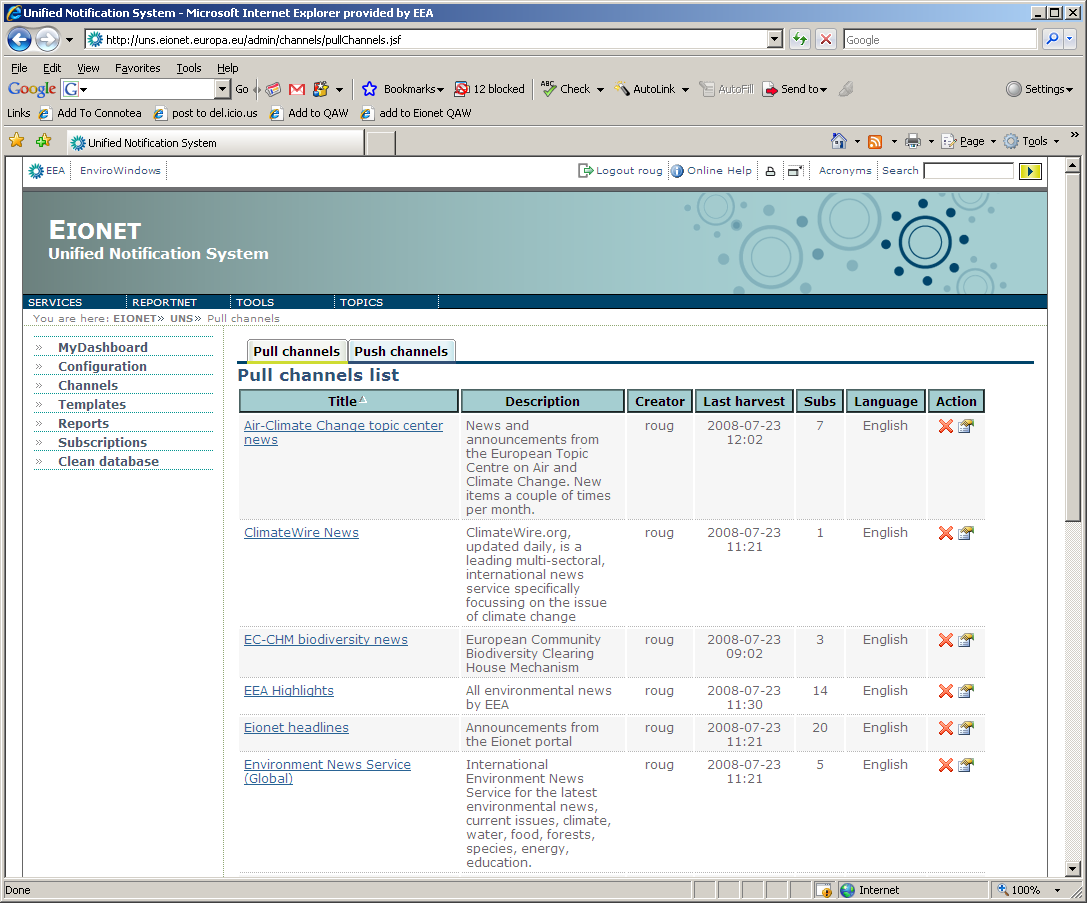
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Channels” button on the navigation bar.

Arriving at the channels section, the user may view the list of existing channels. The “Create” button is found on the bottom of this list.

Figure 4: PULL channels list

STEP 3 – Click on “Create” button

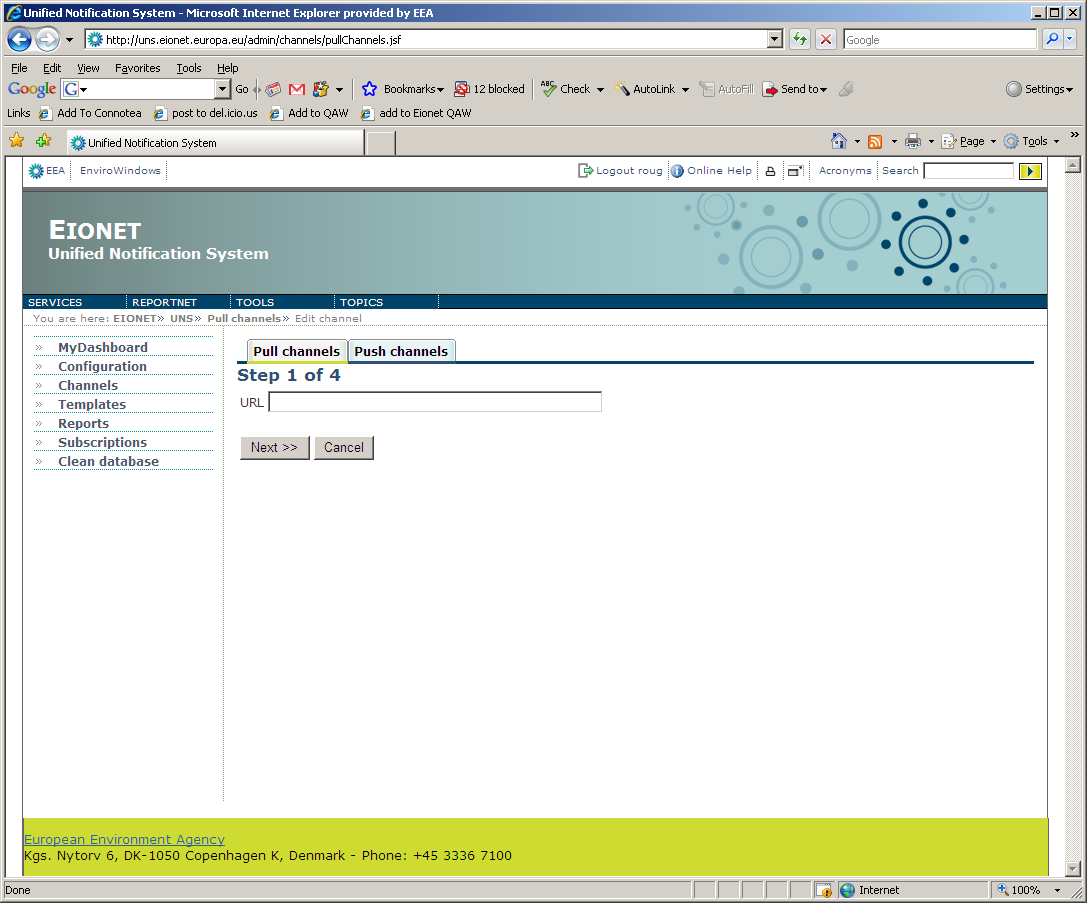
STEP 4 – Insert the required information.

The user has to fill in:

* Channel name: The title of the channel as the end-users will see it
* Channel Description: A short text describing the channel’s context
* URL: Since the channel represents an RSS/RDF feed, this is the address of this feed in the Web.
* Refresh delay: The rate at which the channel’s contents should be refreshed.
* Select notification template
* Select allowed delivery types
* Select roles

In order for the channel to be successfully created, the required information must be filled-in correctly. If all fields are not properly filled-in or the content feed does not exist, then the application will return an appropriate error message.

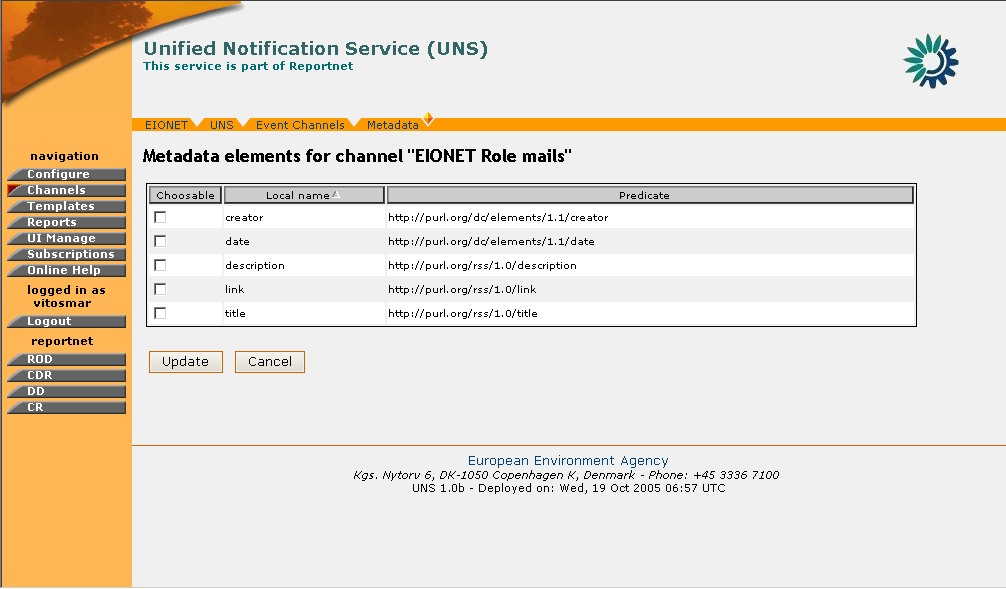
**Important Note**: UNS does not allow the duplication of a channel. If the URL or the name of a new channel already exists, the application will return the appropriate error message.

Figure 5: Channel creation menu

STEP 5 – Select Metadata

After the verification of the information, the user may select the metadata of the channel that Reportnet users may use for filtering the events. An Administrator shall understand the semantics of metadata elements that would like to make available to the user. These elements may not be standard and depend on the event channel feed (see section 2).

In the case that no metadata elements are selected, the user will not be able to filter events arriving for a channel. As a result, the user will receive all events arriving for this channel.

Figure 6: Channel metadata selection

STEP 6 – Create the channel

The channel creation is confirmed and the new channel is visible on the list.

So, in summary:

|  |  |
| --- | --- |
| Step | Create a new channel |
| 1 | Login |
| 2 | Click on “Channels” icon |
| 3 | Click on “Create” button |
| 4 | Insert required information |
| 5 | Select Metadata |
| 6 | Create the channel |

### Delete channel

In order to delete a channel you must be logged in as UNS Administrator.

If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

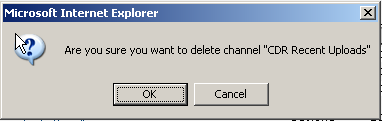
STEP 2 – Click on “Channels” button on the navigation bar.

Arriving at the channels section, the user may view the list of existing channels ().

STEP 3 – Click the  (Delete Channel) icon.

This icon is located at the rightmost column of the channel list

STEP 4 – Confirm channel Deletion

Figure 7: Delete channel confirmation prompt

For safety reasons, the user receives a confirmation prompt for the delete channel action. If the user confirms, then the channel is deleted from the UNS database.

After a channel’s deletion, the channel’s subscribers will receive an automated notification message.

So, in summary:

|  |  |
| --- | --- |
| Step | Delete a channel |
| 1 | Login |
| 2 | Click on “Channels” icon |
| 3 | Click on “Delete Channel” button |
| 4 | Confirm channel deletion |
| 5 | Delete a channel |

### Edit channel

In order to edit a channel you must be logged in as a UNS Administrator.

If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

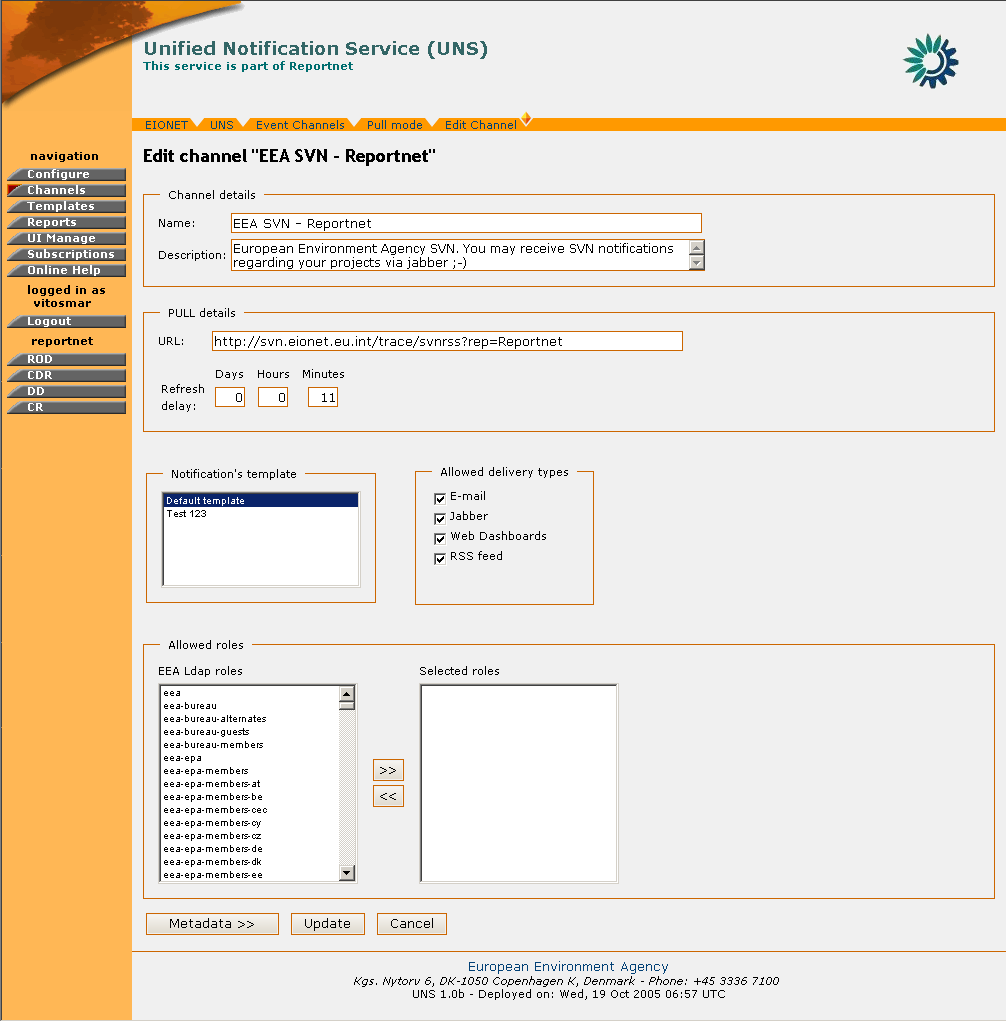
If you are not already logged in, see section 3.

STEP 2 – Click on “Channels” button on the navigation bar.

Arriving at the channels section, the user may view the list of existing channels ().

STEP 3 – Click on any Event channel on the list

By clicking on the channel you will arrive at the “Edit Channel” page.

Figure 8: Edit channel page

STEP 4 – Edit channel fields

In the “Edit Channel” pages you may edit all the channel fields as well as the channel metadata options.

If during this step the channel’s allowed roles are modified, any previously subscribed users that will not have any more access to this channel will receive an automated notification message for this change.

STEP 5 – Push “Update” button

By pressing the “Update” button the channel’s information is updated.

So, in summary:

|  |  |
| --- | --- |
| Step | Edit a channel |
| 1 | Login |
| 2 | Click on “Channels” button |
| 3 | Click on any Event channel on the list |
| 4 | Edit channel fields |
| 5 | Push “Update” button |

## Manage PUSH channels

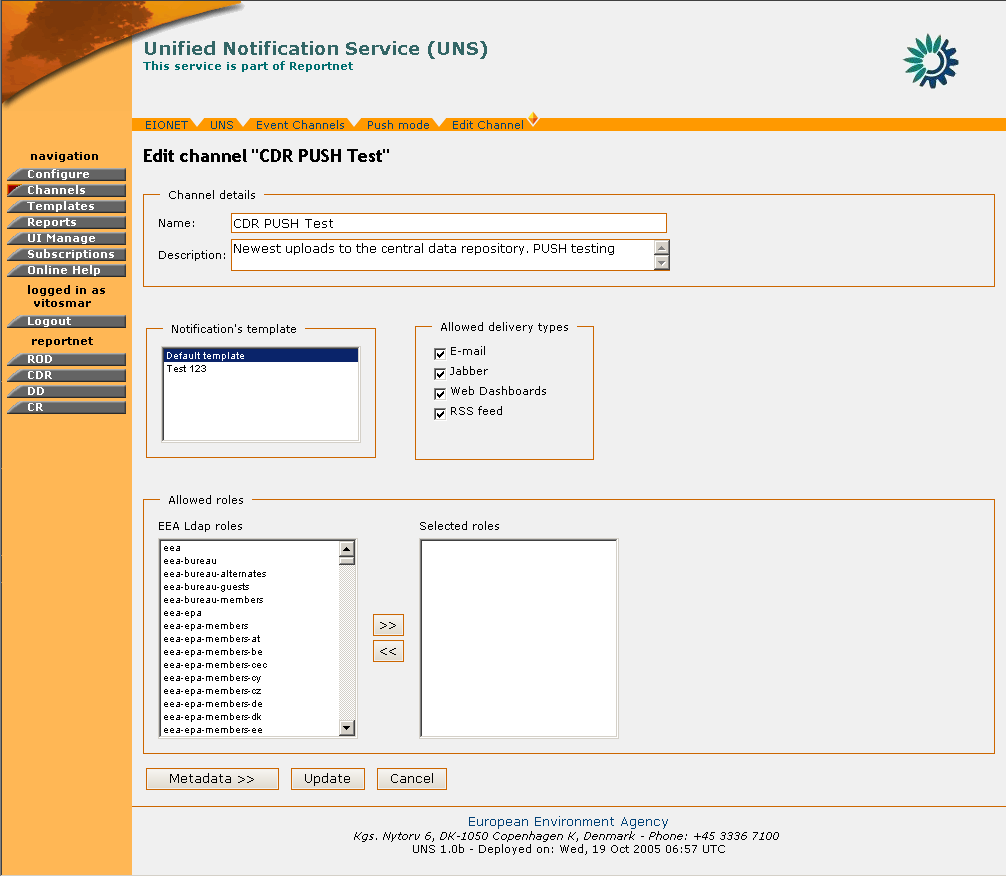
### General principles

Push channels are content services that are managed in UNS. Contrary to the PULL channels, the channels that operate under PUSH mode enable users to insert events in them.

### Manage PUSH channels

PUSH channels cannot be created through the UNS interface by Administrators. They are created by external applications via the XML-RPC API. (See section 6, RPC user features). So an external system supplied by appropriate UNS credentials actually creates the push channels.

But to make this possible the XML-RPC call must be authentcated, normally via a local user created as described in section 4.6.8, Users Administration.

Figure 9: Push channel Edit

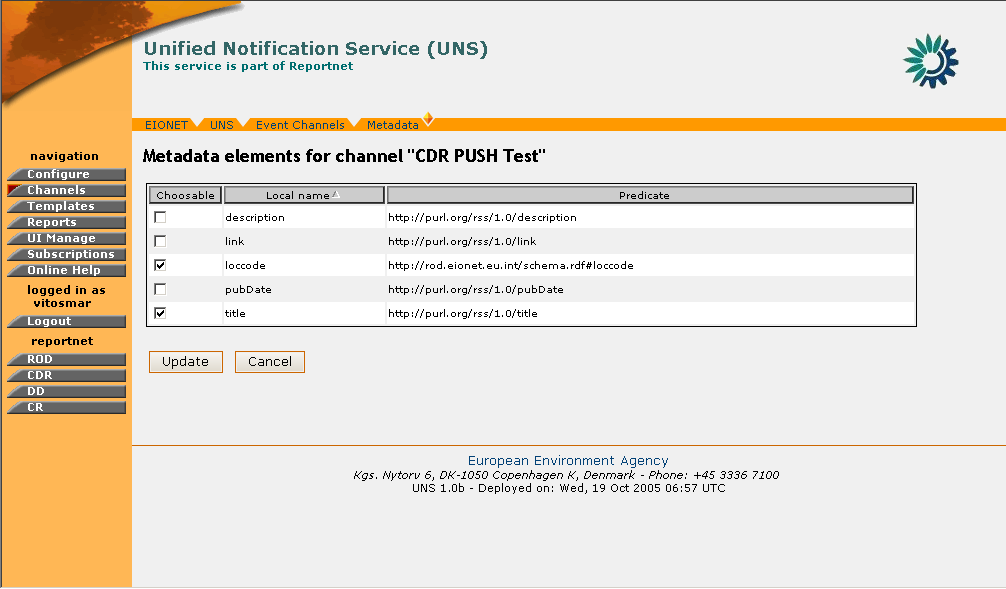
Nevertheless, the UNS administrator is responsible for the management of these channels and can enable or disable them at any time.

More specifically, when the administrator visits the PUSH channels tab under the “Channels” section, a list of the existing PUSH channels is displayed.

Upon the creation of the channel, the name and description are assigned automatically by the system. However, the administrator can edit them and give more meaningful titles. Additionally, the

Administrator can set the channel’s notification template, delivery type and allowed roles parameters. In order to do so, the user has to click on the name of the PUSH channel and arrive at the Channel Edit page.

Similarly to the PULL channels, the PUSH channels also have metadata elements. However, those metadata elements are not defined during the creation of the channel nor can the UNS administrator manage them directly. The PUSH channel metadata are defined at the moment that an initial PUSH of events occurs and subsequently, the UNS administrator may view them and manage them in the same way as in the case of PULL channels.

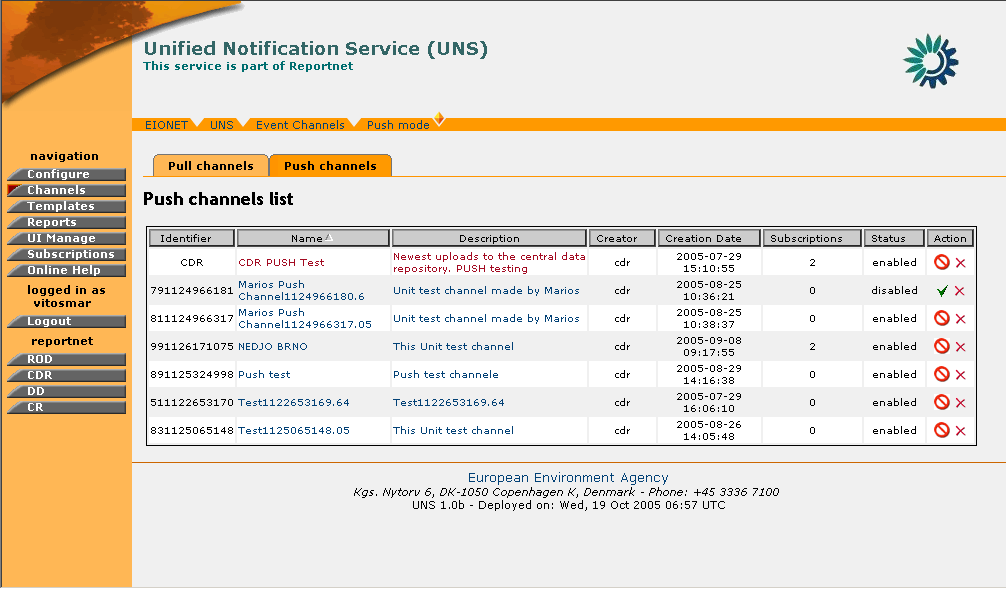
Figure 10: PUSH channel Metadata

As mentioned earlier, the administrator can enable/disable any PUSH channel. This is performed by clicking the corresponding action button on the channel list.

If the channel is under status “enabled”, the user with administration privileges should click the “disable channel” icon  in order to disable it, or if the channel is under status “disabled”, the user should click the “enable channel” icon  to enable the service.

When a channel is under status disabled, then it is not possible for external systems to push events data, neither users to subscribe themselves to that channel.

Furthermore, the user is able to delete an existing PUSH channel by clicking the “delete channel” icon  on the action column.

Figure 11: PUSH channels List

## Notification templates management

### General principles

The format of the notifications that UNS sends to the Reportnet Users is generated according to the notification templates. Notification templates are assigned to specific channels in order for UNS to decide what notification template should be used while generating a notification for a specific subscription.

Each notification generated by a specific channel will be affected by the settings of a template.

For example, concerning the channel *“EEA-SVN Reportnet”*, a normal user that subscribes to the channel *“EEA-SVN Reportnet”* will receive notifications with a format based on the template *“Default Template”.* Notification templates are not used while generating RSS feed or sending events data to the Reportnet Web dashboards.

Three preconfigured notification templates come along with the installation of the UNS. These are:

* Default notification template. Used as a standard event notification template.
* Unsubscribed - channel removed notification. This template delivers a notification message to the subscribers of a channel in case that channel is deleted.
* Unsubscribed - channel access rights changed notification. This template notifies subscribers that due to a change of the channel’s allowed roles they won’t be receiving any more notifications from that channel.

These default notification templates cannot be removed from the system. Nevertheless, they can be customized by UNS administrators. In the notification template, the administrator can personalize the text of the notification as well as the title and the order in which information will be displayed.

Each notification template has two template areas. One for plain text and one for HTML delivery. What the users gets depends on his preferences. If he has set notification to only send plain text, then the plain text template is used, if he prefers HTML email, then *both* templates are used, and the email system will choose for him.

### Placeholders

In order to facilitate the creation and personalization of the notification templates, several “placeholders” are available. These placeholders are inserted where dynamic content [[3]](#footnote-4) is expected. When the notification arrives at the UNS user, these placeholders will contain the full message information.

The currently supported placeholders are:

* $USER: Full name of the user subscribed to receive notifications
* $EVENT.TITLE: Title of the received event
* $EVENT.DATE: Date/time when the Unified notification service received event
* $EVENT.CHANNEL: Name of the channels used for harvesting event
* $EVENT: Harvested Event (with all its data)
* $UNSUBSCRIBE\_LINK: Link that user may use to unsubscribe from channel

For example, if a user needs to create a notification template with a greeting to the receiving user, the body of the template should begin with:

*“* Dear *$USER,*

The UNS has received the following event… *”*

In the same manner, the channel from which the notification message comes from can be indicated in the subject field of the template:

*Subject:* New event from *$EVENT.CHANNEL* channel

### Notification template scripting

The UNS administrative users are able to provide Python scriptlets withing the notification template content. It allows maximum flexibility to the UNS administrative users when defining the notification templates.

An embedded Python script will be marked as follows:

* “<%” represents begin of a Python scriptlet. For writing data into a notification message the administrator must use the print method.
* “%>” represents end of a Python scriptlet
* “<#” implies a print of the result of the Python function. It is terminated with “#>”.

Administrative users are also able to use the predefined placeholders described above, but not within Python scriptlets.

The following example presents a template that contains embedded Python code:

Dear <# subscription['user']['fullName'] #>

Event retrieved through channel from $EVENT.CHANNEL contains the following metadata:

<%

for key,values in metadata\_dict.items():

print '%s %s' % (key,",".join(values))

%>

You may unsubscribe yourself from the “$EVENT.CHANNEL” channel by using the following link: $UNSUBSCRIBE\_LINK

Best regards,

European Environment Agency

Two UNS objects are available for direct manipulation inside the Python scriptlets as follows:

event Dictionary containing the event's details and meta data. It has the following attributes: date, title, metadata. Metadata is a dictionary using the predicate URL as the key. There is a known bug in the metadata. If there are more than one value for a predicate only the last one is stored.

subscription Dictionary containing user subscription details. It has the two attributes: channel, which is a reference to the channel object, and user, which is a reference to the user object.

metadata\_dict Metadata\_dict is a dictionary using the predicate URL as the key. The value is always a list, even if there is only one item in it. Therefore always use “, “.join(values) or similar.

metadata\_list A list of predicate, value pairs. Unsorted.

Indirectly from the subscription object you can reference to the following object:

channel Dictionary containing the channel information. If has the following attribute: title

user Dictionary containing user details. It has the following attributes: fullName, externalId.

For more details see the source code file: src/main/python/UNS/Notifications/\_\_init\_\_.py

### Create new notification template

In order to create a Notification template you must be logged in as UNS Administrator.

If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Templates” button on the navigation bar.

Arriving at the Templates section, the user can view the list of existing notification templates.

STEP 3 – Click on “Create” button

In order to create a new notification template, you must click on the “Create” button and arrive at the template creation page.

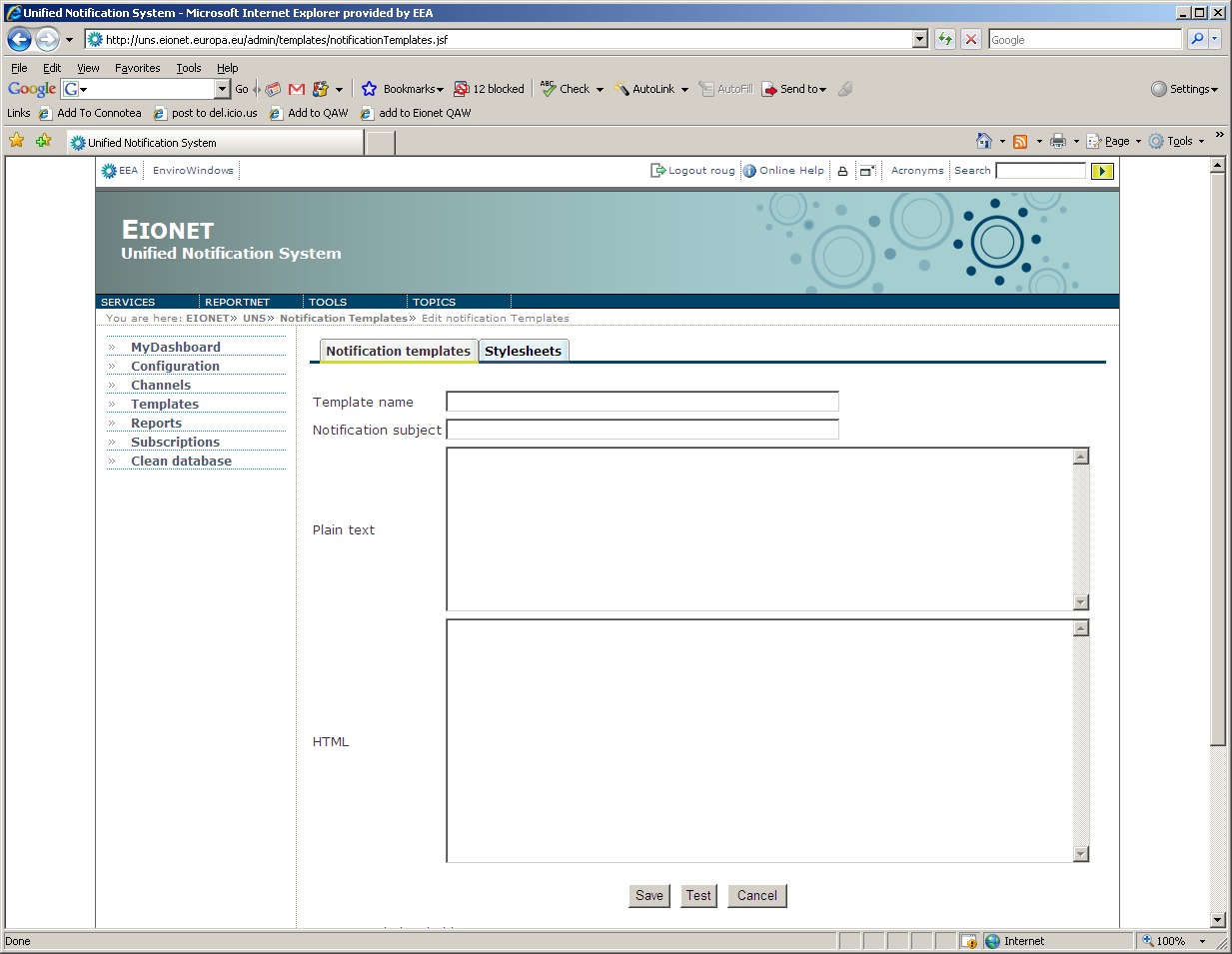
STEP 4 – Fill-in required information

The information that needs to be inserted is: Template name, Message subject and Message Text as both plain text and HTML delivery.

The message subject and the template fields are those which will be displayed at the user for every notification, while the template name is used solely for the identification of the template by the administrator.

Placeholders such as $USER, $EVENT etc. can be inserted to represent dynamic content which is decided upon the delivery of the notification. For more information on placeholders see section 4.4.1.

For the HTML message, remember to enter all the HTML code, i.e. <html>, <head> and <body>.

Figure 12: Notification template creation page

STEP 5 – Create Notification template

After inserting all the required information, press the “Update” button for the new Notification template to be created. This new template will be visible in the Notification template list.

In summary:

| Step | Create new Notification template |
| --- | --- |
| 1 | Login |
| 2 | Click on “Templates” button |
| 3 | Click on “Create” button |
| 4 | Fill-in required information |
| 5 | Create Notification template |

### Notification template Editing

In order to edit a Notification template you must be logged in as UNS Administrator.

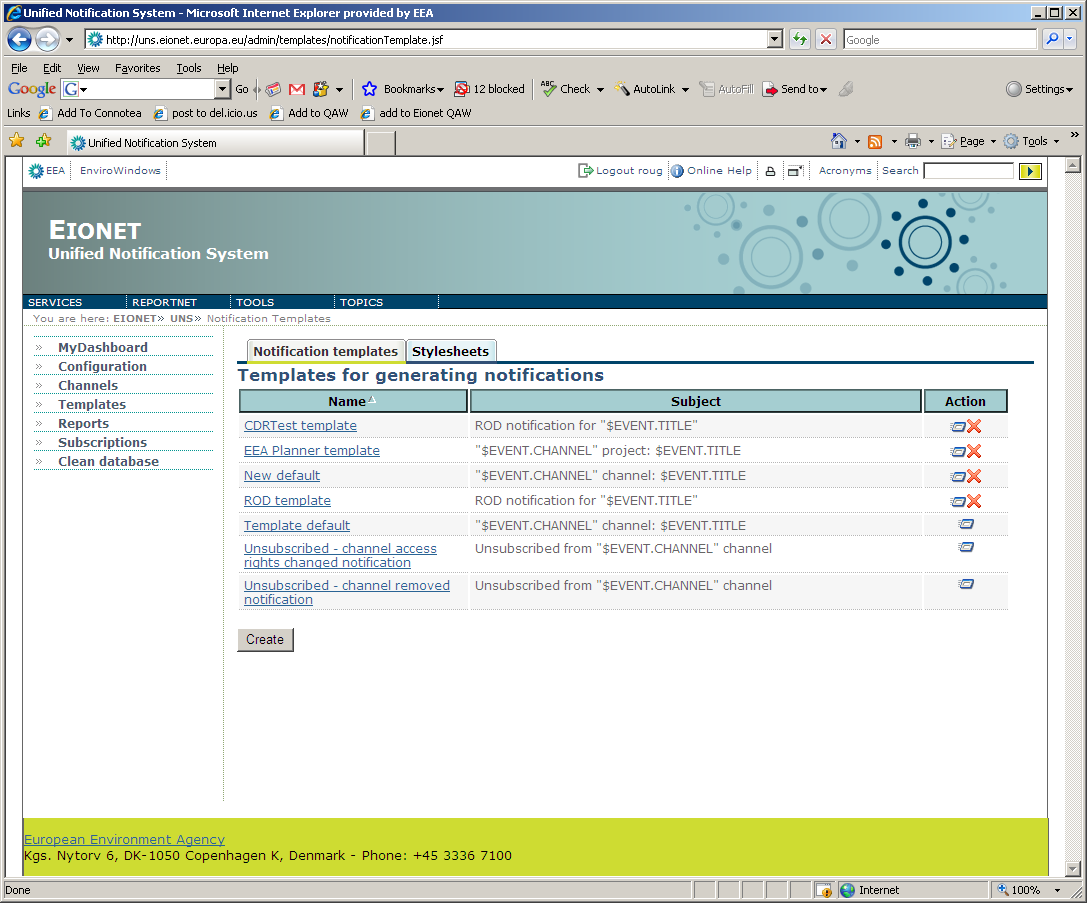
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Templates” button on the navigation bar.

Arriving at the Templates section, the user can view the list of existing notification templates.

Figure 13: Notification template list

STEP 3 – Select template to edit

Click on the name of an existing template in order to edit it. This will take you to the “Edit Template” page where the current template information is displayed.

STEP 4 – Edit Notification template

Any of the Notification template fields can be edited. You may perform any changes on the existing template data and decide to save changes.

STEP 5 – Update Notification template

You must press the “Update” button in order for the changes to take effect.

In summary:

| Step | Edit Notification template |
| --- | --- |
| 1 | Login |
| 2 | Click on “Templates” button |
| 3 | Select template to edit |
| 4 | Edit Notification template |
| 5 | Update Notification template |

### Notification template Deletion

In order to delete a Notification template you must be logged in as UNS Administrator.

\* **Important note**: A Notification template cannot be deleted if it is used by a channel.

If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Templates” button on the navigation bar.

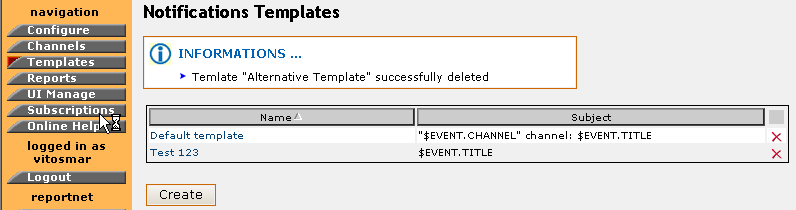
Arriving at the Templates section, the user can view the list of existing notification templates.

STEP 3 – Select template to delete

Tick the box of the template you wish to delete. Then press the “Delete” button on the bottom of the screen.

STEP 4 – Delete Notification template

If the deletion was successful, the appropriate message will be displayed and the Notification template list will be updated respectively.

Figure 14: Information message after notification template deletion

In summary:

| Step | Edit Notification template |
| --- | --- |
| 1 | Login |
| 2 | Click on “Templates” button |
| 3 | Select template to delete |
| 4 | Delete Notification template |

## Management and monitoring of notifications

### General principles

The UNS gives the possibility to the administrator to get feedback on the system performance in terms of the amount of notifications sent for a given time period as well as a report of notifications which failed to be delivered.

This is realized through the “Reports” section, where privileged users can generate such reports.

The system allows the administrator to view reports for any given time period as well as to define the report on specific users and channels.

### Notifications throughput report

In order to generate a Notifications Throughput report, you must be logged in as UNS Administrator.

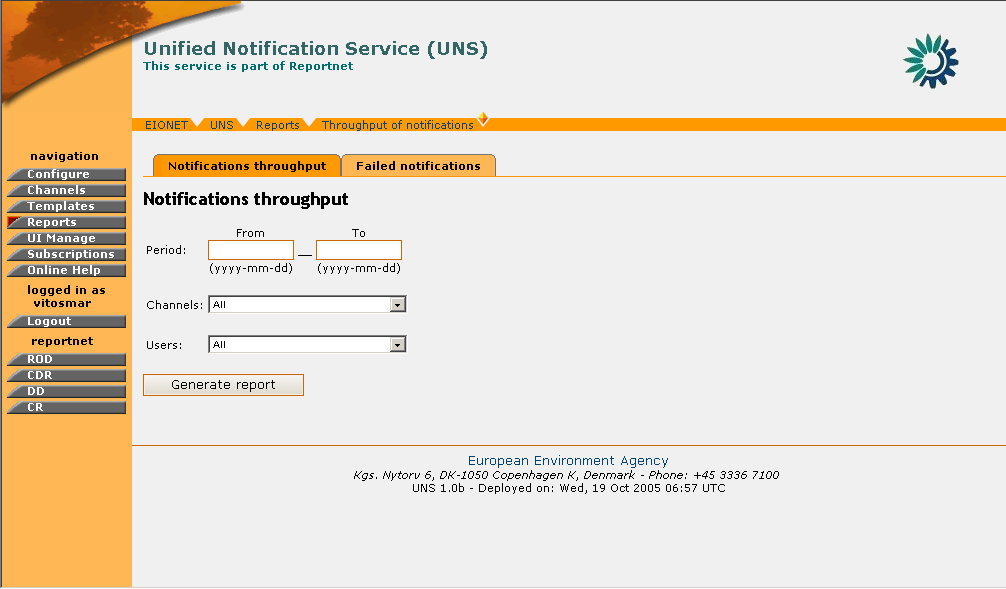
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Reports” button on the navigation bar.

In order to create a Notifications Throughput Report, the “Notifications throughput” tag must be selected.

Figure 15: Notifications Throughput page

STEP 3 – Fill-in the Notifications Throughput Form

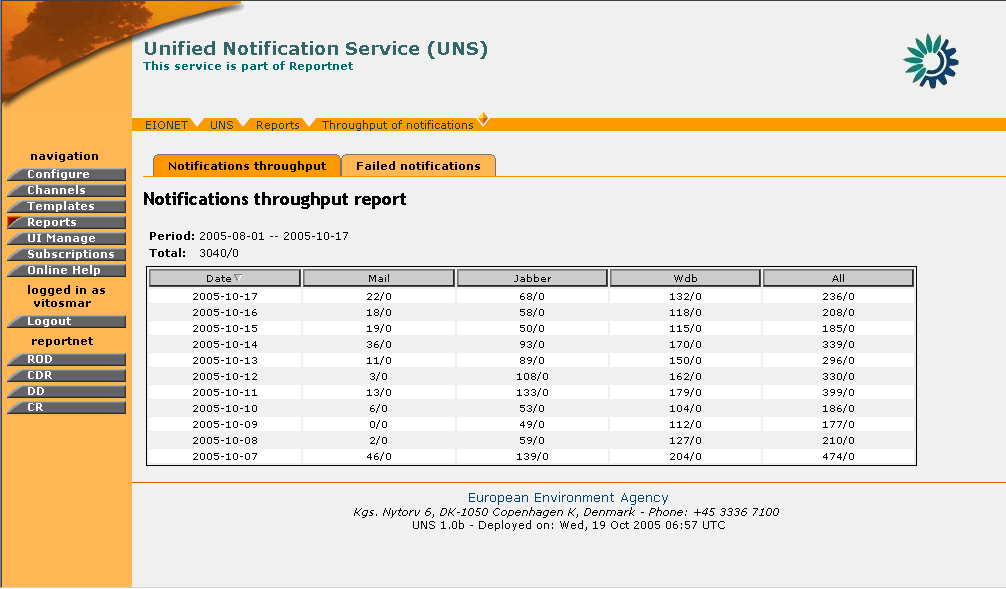
The user must fill-in the corresponding form depending on the data he/she wishes to view.

The period field is used to specify the time interval which is going to be taken into consideration for the report, while the Channel and User fields allow the user to narrow down the report results to a specific channel, or a specific user or both.

The option “All” if selected for both Channel/User fields, will denote the generation of a report listing all notifications sent in all channels for the specified time period.

STEP 4 – Receive Notifications Throughput Report

After filling-in the form, the user must select the “Generate Report” button. When this is done, the results of the user’s search will be displayed.

Figure 16: Sample Notifications Throughput Report

The report is comprised of the following parts:

* At the top, a summary of the search criteria is presented, as well as the Total number of successful Notifications sent for the given time period.
* The analytical table, where the user may observe the amount of notifications throughput in a daily basis, split by the type of notification sent (Mail, Jabber, WDB, all)
* For any Notification Throughput data, two numbers are returned which are split by a forward slash “/”. The part on the left of the slash represents the number of successful notifications sent while the right part represents the number of failed notifications.

In summary:

|  |  |
| --- | --- |
| Step | Edit Notification template |
| 1 | Login |
| 2 | Click on “Reports” button |
| 3 | Fill-in Notifications Throughput Form |
| 4 | Receive Notifications Throughput Report |

### Failed Notifications Report

In order to view a Failed Notifications report, you must be logged in as UNS Administrator.

If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

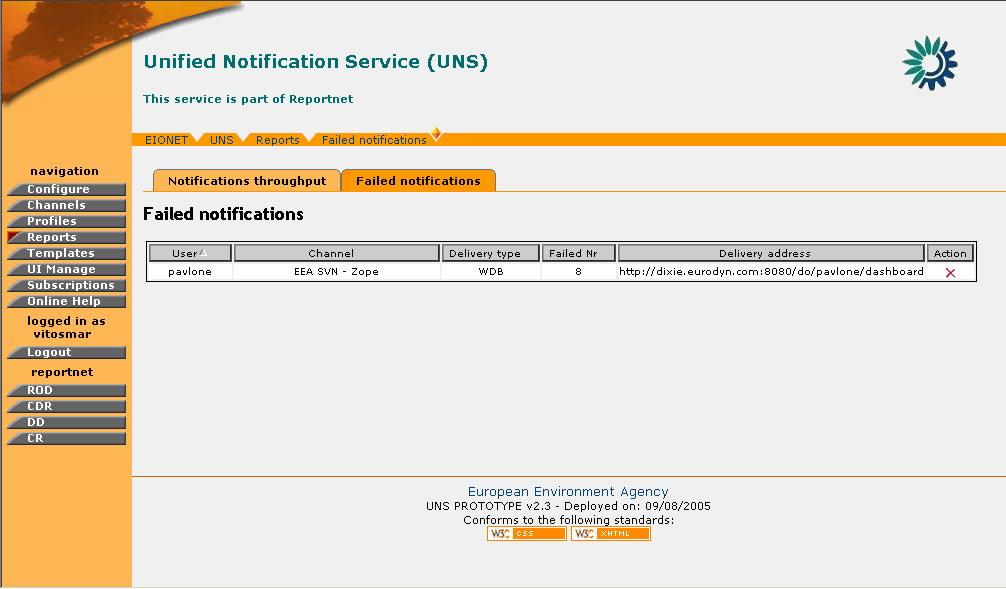
If you are not already logged in, see section 3.

STEP 2 – Click on “Reports” button on the navigation bar.

In order to view the notifications which failed to deliver, the “Failed Notifications” tag must be selected.

STEP 3 – View Failed Notifications Report

By selecting the “Failed notifications” tag, the following screen will be displayed:

Figure 17: Sample Failed Notifications Report

This report shows all failed notifications (if any) recorded in the system. Hence, there is no time interval specification like in Notifications Throughput reports.

The report displays the user for which these notifications were intended, the channel from which they originated the delivery type (email, WDB, Jabber), the delivery address which failed and the number of failed notifications.

STEP 4 – Unsubscribe user from channel

If any failed notifications exist for a user, the administrator may remove the user’s subscription that causes these notifications to appear. In this way, the flood of the system with invalid subscriptions is prevented.

The administrator can perform this directly through the failed notifications report, by selecting the button in the “Action” column.

In summary:

|  |  |
| --- | --- |
| Step | View Failed Notifications Report |
| 1 | Login |
| 2 | Click on “Reports” button |
| 3 | View Failed Notifications Report |
| 4 | Unsubscribe User from channel |

## UNS configuration

### General principles

The UNS application gives to the administrator possibility to configure and modify several application parameters.

Figure 18: UNS Configuration Options

More specifically, the user can configure the following:

* Mail. The user can modify the SMTP and POP3 server settings.
* LDAP. The user can modify the LDAP server parameters. Through this server UNS users are authenticated.
* Database. Through this option, the user will be able to modify the UNS database settings.
* Jabber. The user can modify the parameters of the server to which any Jabber notifications will be delivered to.
* Harvester. The Harvester daemon is responsible for fetching new items from the channels. The user may start/stop the Harvester or change the harvesting periodicity.
* Notificator. The Notificator daemon is responsible for delivering harvested events as notifications to Reportnet users and for checking of failed e-mail notifications. The user may start/stop the Notificator or change the notification sending periodicity.

### Mail Parameters

In order to edit the Mail parameters, you must be logged in as UNS Administrator.

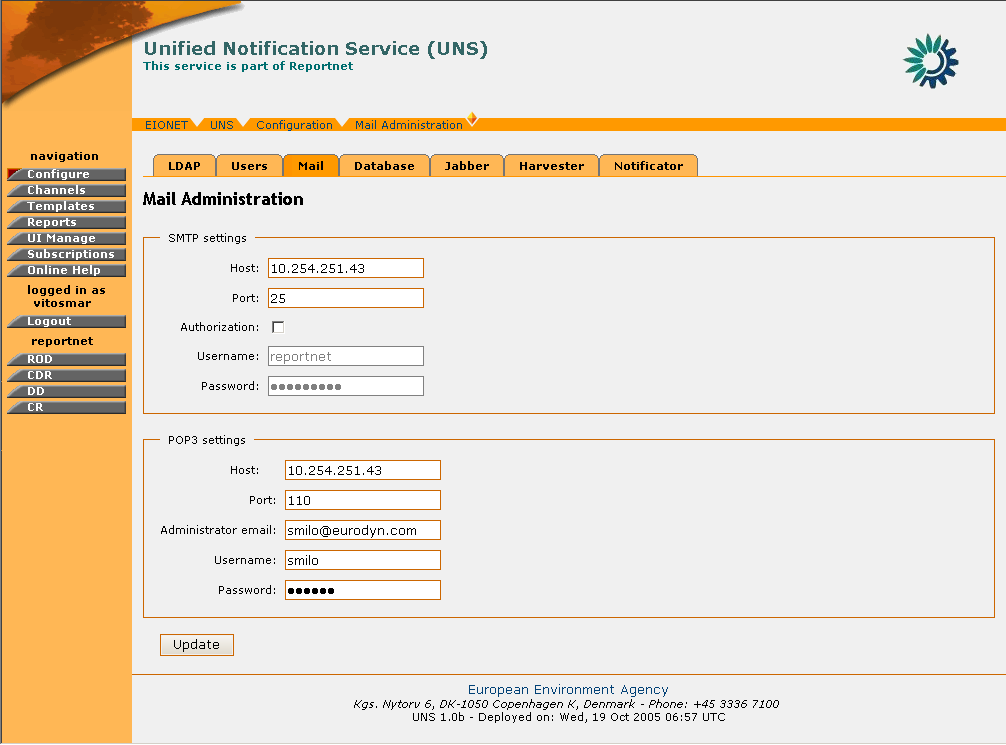
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to edit the Mail options, the “Mail” tag must be selected.

Figure 19: Mail Parameters page

STEP 3 – Edit values for SMTP/POP3 Servers

For each of the servers, the following fields are editable:

* Host: The host name or IP address of the SMTP/POP3 servers.
* Port: The network port at which the server is listening
* Username/Password (if required): The credentials that are used to access the server
* Administrator e-mail (POP3 only): The email address of the server’s administrator.

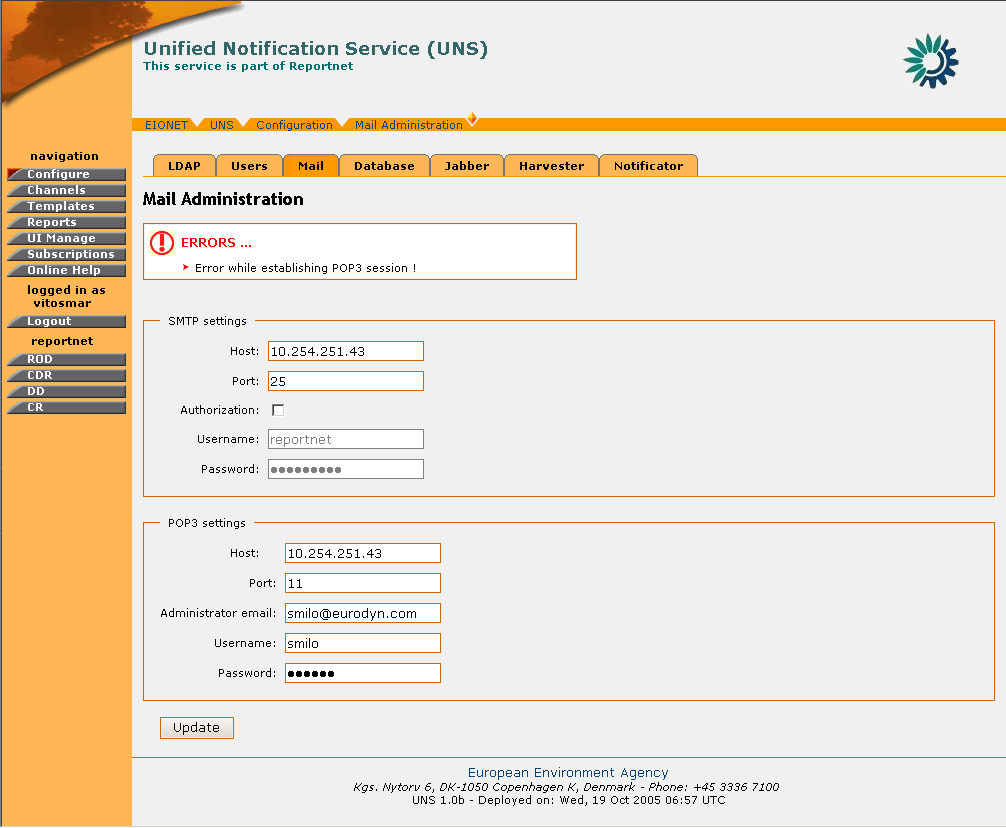
The user can edit any of the server settings available. Changing the SMTP server settings will affect the server through which e-mail notifications are sent.

The POP3 server is primarily used for retrieval of failed mail notifications. The Administrator e-mail address is used by the UNS to identify itself as sender of the notifications and to identify the inbox on the server where it shall look for returned notifications.

\* **Important note**: A Mail parameter cannot be changed if Notifications daemon is running. An administrative user needs to stop Notificator daemon before trying to change mail parameters.

STEP 4 – Update Configuration

In order for any changes to take effect the user must select the “Update” button. If the operation is successful, the appropriate information message will be displayed. However, if any of the new settings are not valid (wrong address of server/port, wrong username/­password) the application will not allow the configuration update and return the appropriate error message.

Figure 20: Error message in Mail parameters page

Similar error messages will also be returned in case the user tries to change the Mail configuration while the Notificator Daemon is running. In such cases the Notificator must first be shut down (see section 4.6.7).

In summary:

|  |  |
| --- | --- |
| Step | Edit Mail Parameters |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Edit values for SMTP/POP3 Servers |
| 4 | Update Configuration |

### LDAP Parameters

In order to edit the LDAP parameters, you must be logged in as UNS Administrator.

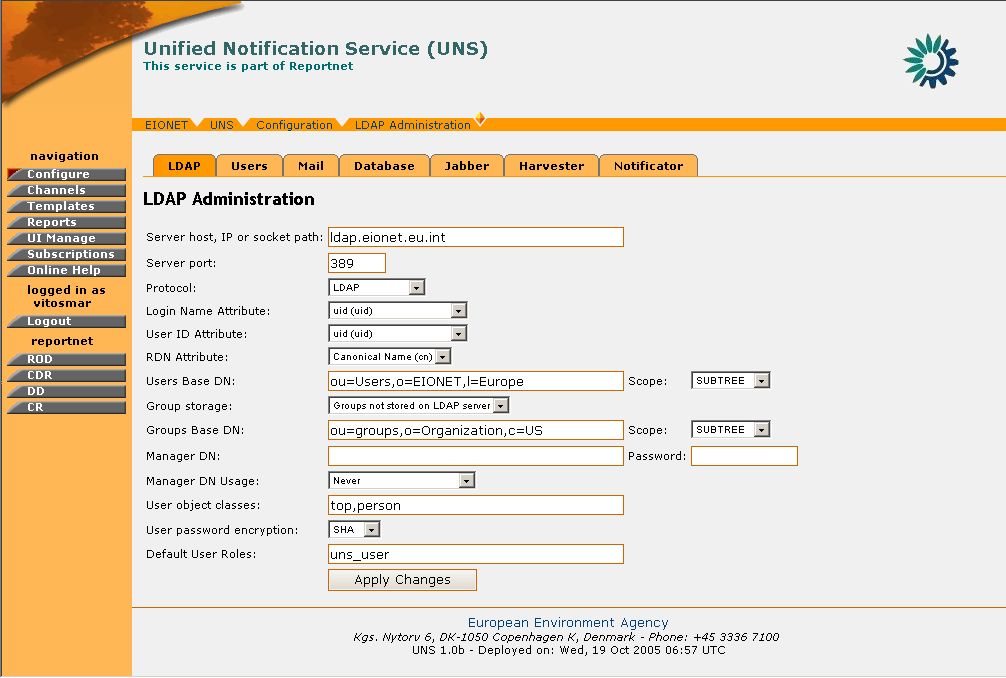
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to edit the LDAP options, the “LDAP” tag must be selected.

Figure 21: LDAP Parameters page

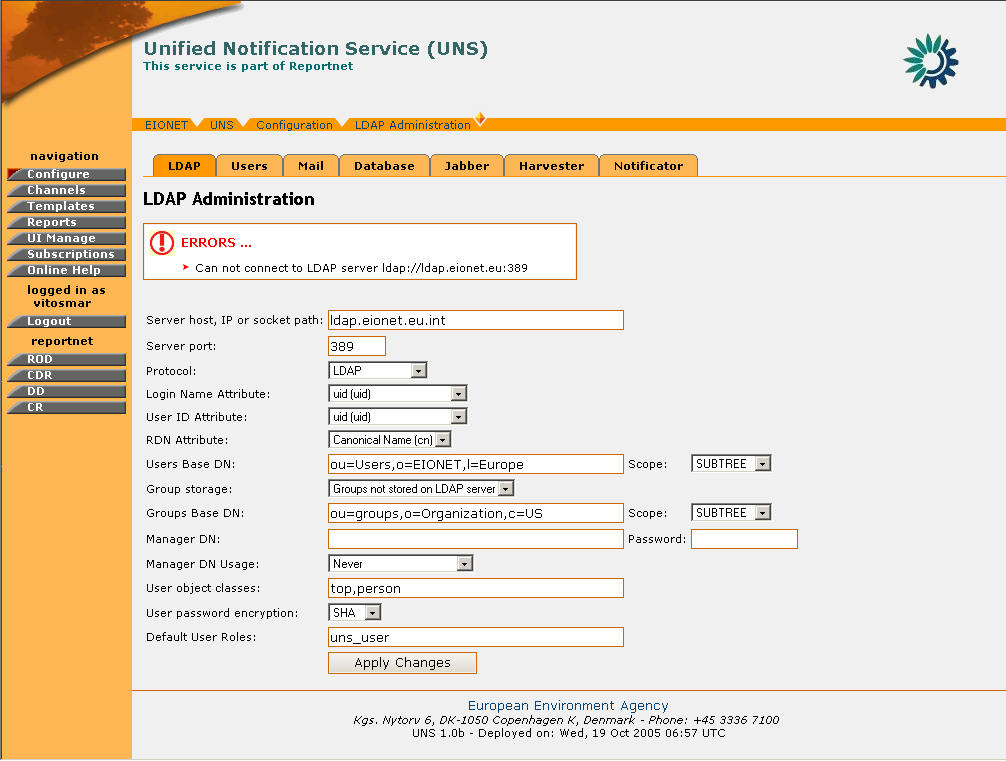
STEP 3 – Edit values for LDAP server

The user can edit any of the server parameters. Changing the LDAP server parameters will affect the UNS user authentication. The following fields are editable:

* Server host, IP or socket path: The hostname or IP address for the LDAP server.
* Server port: The port that the LDAP server is listening on. By default, an LDAP server listens on port 389.
* Protocol: Select whether to use standard LDAP, LDAP over SSL or LDAP over IPC.
* Login Name Attribute: The LDAP record attribute used as the username. The list of default choices can be changed by adding attributes on the LDAP Schema tab in the LDAPUserFolder product instance.
* User ID Attribute: The LDAP record attribute used as the user ID. The list of default choices that can be changed by adding attributes on the LDAP Schema tab in the “*LDAP UserFolder*” product instance.
* RDN Attribute: The RDN attribute (Relative Distinguished Name) is the name of the LDAP attribute used as the first component of the full DN (Distinguished Name).
* Users Base DN: The DN for the branch of the LDAP database that contains user records.
* Scope: Choose the depth for all searches from the user search base DN.
* Group storage: Choose where to store the group (a.k.a. Role) information for users.
* Groups Base DN: The DN for the branch of the LDAP database that contains group records. These group records are of the LDAP class "groupOfUniqueNames" and the entry CN attribute constitutes the group name.
* Scope: Choose the depth for all searches from the group search base dn. If you have chosen to store groups inside the user folder itself this setting will be disregarded.
* Manager DN and password: Under normal operation if no separate Manager DN is provided, the UNS will use the current user's DN and password to authenticate against the LDAP server. If a Manager DN and password are given, these will be used instead.
* Manager DN usage: Specify how the Manager DN (if it has been provided) will be used.   
  The option “Never” means that will never apply this DN. If no Manager DN is specified then this is the default value. The option “Always” means that the Manager DN is used to bind for every single operation on the LDAP server.   
  The option “For login data lookup only” uses the Manager DN upon user login when the user itself has not been instantiated yet and thus the user's DN is not yet known. Once the user has been instantiated its DN and password are used for binding.
* User object classes : Comma-separated list of object classes for user records. Any new user record created through the “*LDAPUserFolder*” will carry this list of object classes as its objectClass attribute.
* User password encryption : This dropdown specifies the encryption scheme used to encrypt a user record “*UserPassword*” attribute. This scheme is applied to the plaintext password when a user edits the password or when a new user is created in LDAPUserFolder product. Check your LDAP server to see which encryption schemes it supports, pretty much every server can at least do "crypt" and "SHA".
* Default User Roles: All users authenticated from the LDAP tree will be given the roles user put into this comma-delimited list.

STEP 4 – Update Configuration

In order for any changes to take effect the user must select the “Apply Changes” button. If the operation is successful, the appropriate information message will be displayed. However, if any of the new settings are not valid (wrong address of server/port, wrong user base) the application will not allow the configuration update and will return the appropriate error message.

Figure 22: Error message while saving invalid LDAP settings

In summary:

|  |  |
| --- | --- |
| Step | Edit LDAP Parameters |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Edit values for LDAP Servers |
| 4 | Update Configuration |

### Database Parameters

In order to edit the Database parameters, you must be logged in as UNS Administrator.

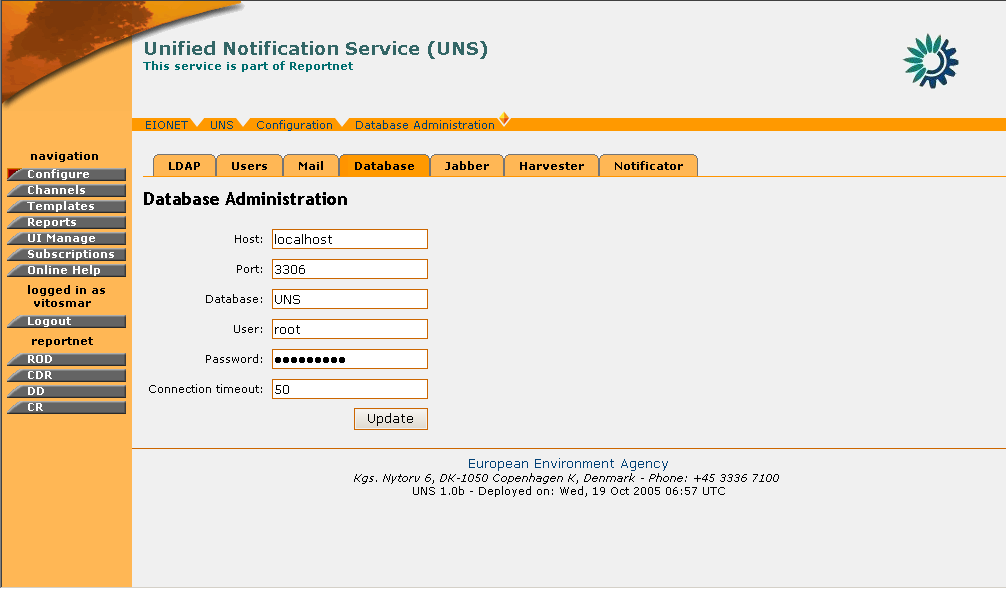
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to edit the LDAP options, the “Database” tag must be selected.

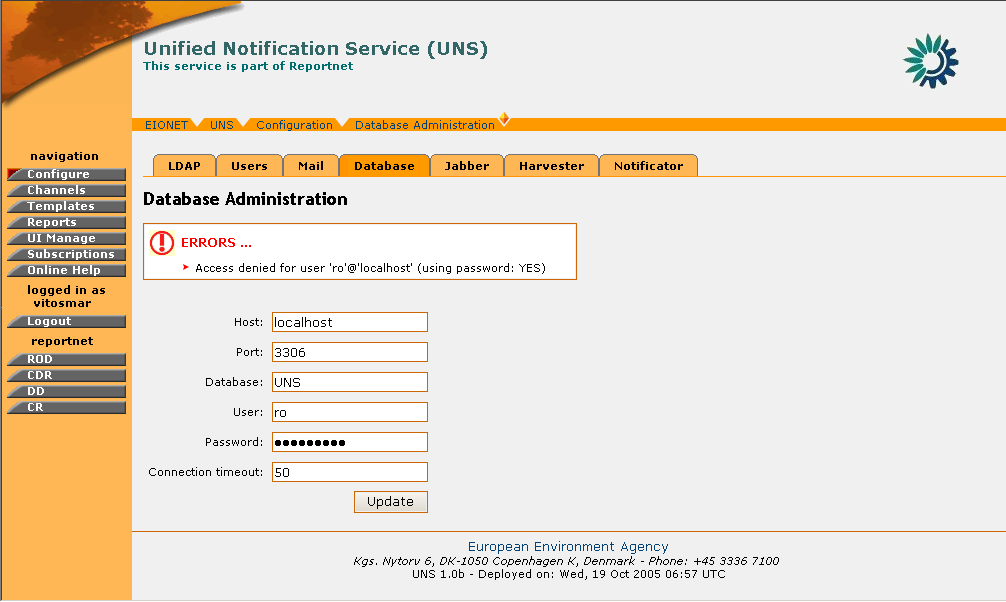
Figure 23: Database Administration page

STEP 3 – Edit values for UNS Database

The user can edit any of the database connection parameters. Changing these parameters will affect the data the UNS application accesses.

STEP 4 – Update Configuration

In order for any changes to take effect the user must select the “Update” button. If the operation is successful, the appropriate information message will be displayed. However, if any of the new settings are not valid (wrong address of database/port, wrong username/password) the application will not allow the configuration update and will return the appropriate error message.

Figure 24: Error message while saving invalid database connection settings

Similar error messages will also be returned in case the user tries to change the Database configuration while the Harvester daemon is running. In such cases the Harvester must first be shut down (see section 4.6.6).

**Important note**: A Database parameter cannot be changed if Notifications and/or Harvester daemon are running. An administrative user needs to stop those daemons before trying to change database parameters.

In summary:

|  |  |
| --- | --- |
| Step | Edit Database Parameters |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Edit values for UNS Database |
| 4 | Update Configuration |

### Jabber Parameters

In order to edit the Jabber parameters, you must be logged in as UNS Administrator.

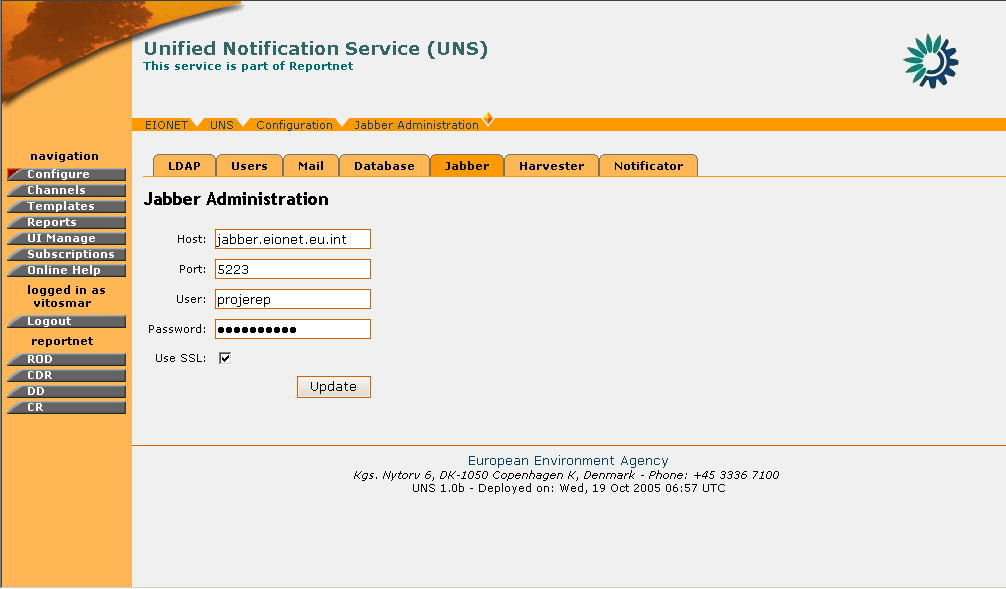
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3 (Authenticate User Account).

STEP 2 – Click on “Configure” button on the navigation bar.

In order to edit the Jabber options, the “Jabber” tag must be selected.

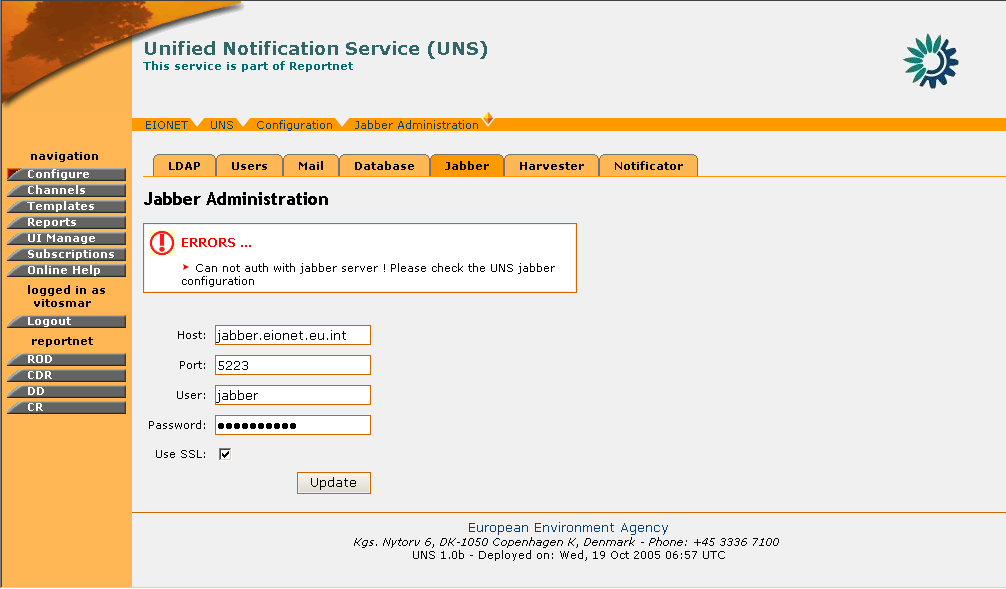
Figure 25: Jabber settings page

STEP 3 – Edit values for Jabber Server

The user can edit any of the Jabber server connection parameters. Changing these parameters will affect the server to which the Jabber notification messages will be delivered to.

STEP 4 – Update Configuration

In order for any changes to take effect the user must select the “Update” button. If the operation is successful, the appropriate information message will be displayed. However, if any of the new settings are not valid (wrong server hostname/port, wrong username/password) the application will not allow the configuration update and will return the appropriate error message.

Figure 26: Error message while saving invalid Jabber server settings

Similar error messages will also be returned in case the user tries to change the Jabber configuration while the Notificator daemon is running. In such cases the Notificator must first be shut down (see section 4.6.7).

**Important note**: A Jabber parameter cannot be changed if Notifications daemon is running. An administrative user needs to stop Notifications daemon before trying to change jabber parameters.

In summary:

|  |  |
| --- | --- |
| Step | Edit Jabber Parameters |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Edit values for Jabber Server |
| 4 | Update Configuration |

### Harvester Configuration

The Harvester daemon is the module that browses through the content channels in order to check for new events. In order to edit the Harvester daemon settings, you must be logged in as UNS Administrator.

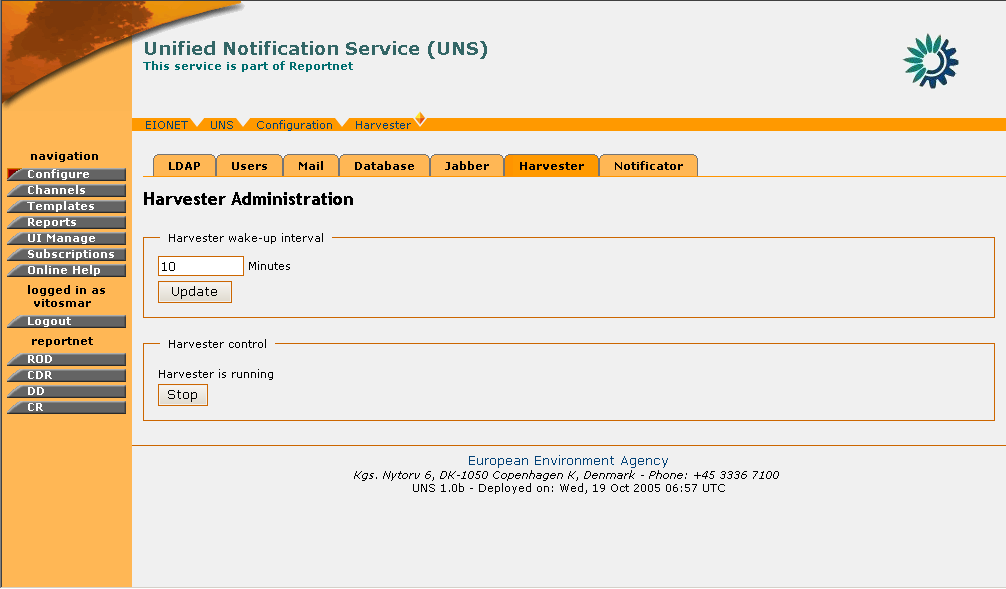
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to edit the Harvester options, the “Harvester” tag must be selected.

Figure 27: Harvester Daemon administration page

STEP 3 – Edit values for Harvester Daemon

The user the following options:

* Set the time interval for which the Harvester will search channels for new events
* Start/Stop the Harvester

STEP 4 – Update Configuration

In order for any changes in the Harvester innovation interval to take effect the user must select the “Update” button.

However, if the user wishes to Start/Stop the Harvester, the corresponding Start/Stop button on the Harvester control box needs to be selected. Upon the selection of the button the screen will be refreshed and the new Harvester status will be displayed.

In case of error, the appropriate error message will be returned.

In summary:

|  |  |
| --- | --- |
| Step | Edit Harvester Configuration |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Edit values for Harvester Daemon |
| 4 | Update Configuration |

### Notificator Configuration

The Notificator daemon is the module that sends notifications for harvested events to UNS users.

In order to edit the Notificator daemon settings, you must be logged in as UNS Administrator.

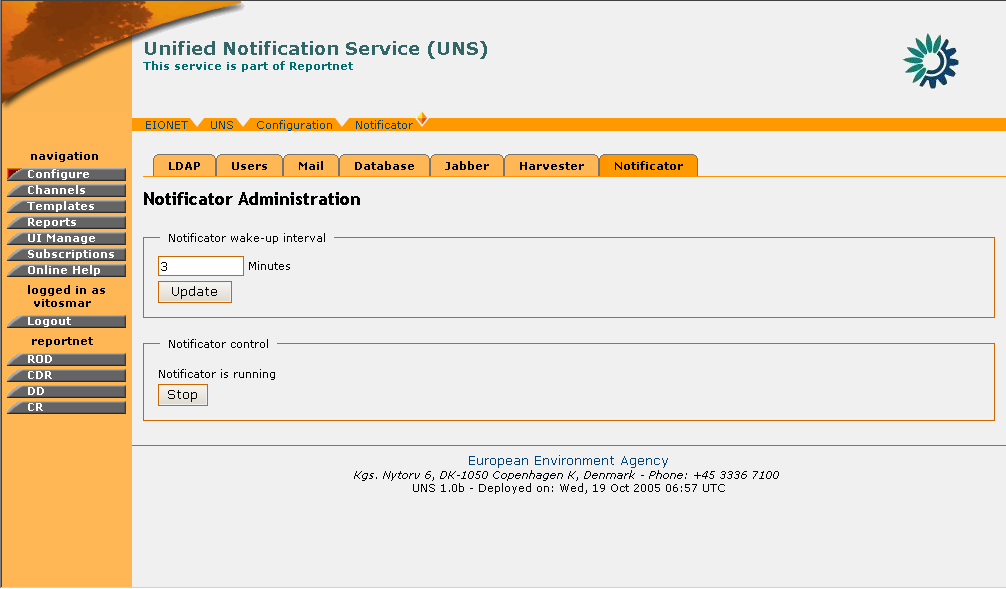
If you have administrator privileges, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to edit the Notificator options, the “Notificator” tag must be selected.

Figure 28: Notification Daemon administration page

STEP 3 – Edit values for Notificator Daemon

The user the following options:

* Set the time interval for which the Notificator will send new events to UNS users
* Start/Stop the Notificator

STEP 4 – Update Configuration

In order for any changes in the Notificator innovation interval to take effect the user must select the “Update” button.

However, if the user wishes to Start/Stop the Notificator, the corresponding Start/Stop button on the ‘Notificator control’ box needs to be selected. Upon the selection of the button the screen will be refreshed and the new Notificator status will be displayed.

In case of error, the appropriate error message will be returned.

In summary:

| Step | Edit Notificator Configuration |
| --- | --- |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Edit values for Notificator Daemon |
| 4 | Update Configuration |

### Users Administration

The Users Administrator tool is used to manage and create users on the UNS.

The following user classes have access to the Users Administation:

1. UNS Administrator
2. Zope Manager

The difference between the two is that the Zope Manager can manage both UNS Administrators and UNS\_rpc users, while the UNS Administrator can manage only UNS\_rpc users.

Each of those user classes can create or delete users from the UNS.

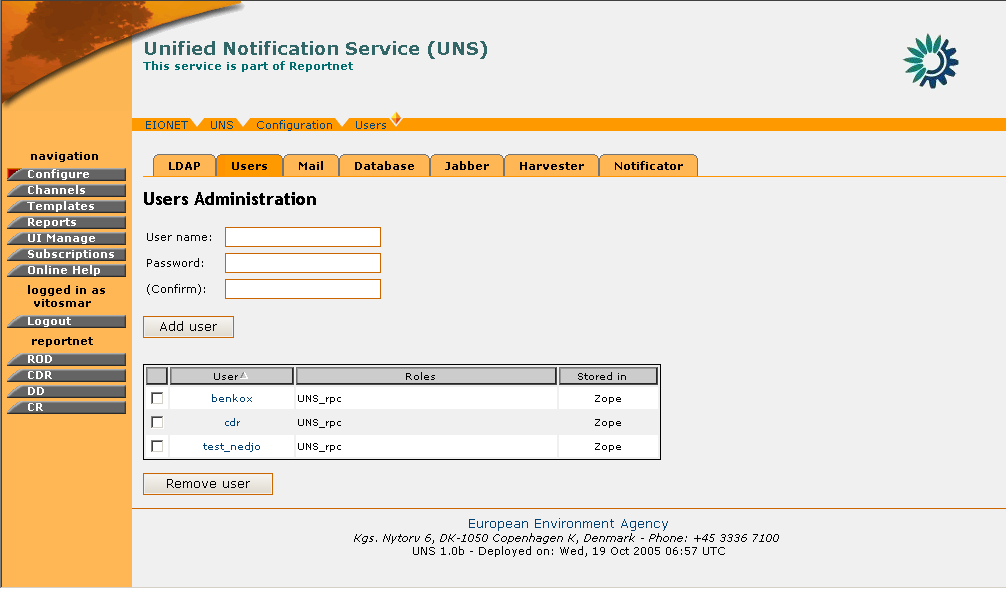
In order to create a user, the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to create a new user, the “Users” tag must be selected.

Figure 29: Users Administration

STEP 3 – Enter required information

The user has to enter the following fields:

* Username
* Password
* User class (available only for the Zope Manager),

\* **Important Note:** If the user name specified, has already been assigned to a user either located on Zope or located on the LDAP server, the application will return an error with the appropriate message.

STEP 4 – Create User

Push the “Create User” button in order for the new user to be created. The new user will become visible on the corresponding table.

In summary:

| Step | Create New User |
| --- | --- |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Enter Required Information |
| 4 | Create user |

In order to remove a user, the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Configure” button on the navigation bar.

In order to create a new user, the “Users” tag must be selected.

STEP 3 – Select user(s) to be removed

Select the desired user(s) to be removed by ticking on the appropriate box. Multiple users can be selected

STEP 4 – Remove User

Push the “Remove User” button in order for the selected user(s) to be deleted.

In summary:

| Step | Remove User |
| --- | --- |
| 1 | Login |
| 2 | Click on “Configure” button |
| 3 | Select user(s) to be removed |
| 4 | Remove user |

# UNS User FEATURES

## General principles

The user module section describes the set of UNS operations and functions which are available to simple non-administrator users. This set of functions allows Reportnet users to manage their notification profile information. More precisely, it allows registered users to specify what type of notifications they would like to receive and how they like to receive it.

This set of functions is comprised of the following:

* Subscribe to an event channel
* Unsubscribe from an event channel
* Edit Subscription

## Notification profile management

### Setting notification preferences and vacation flag

The notification preferences of any Reportnet user can be set by logging-in the UNS and accessing the “My preferences” tab.

From this page, the user may define the delivery addresses of the messages destined for the Web Dashboards and Jabber services. In addition, the user may enable or disable the vacation flag.

#### Setting delivery addresses

In order for each user to set the delivery addresses for the Jabber and WDB services, the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3 (Authenticate User Account).

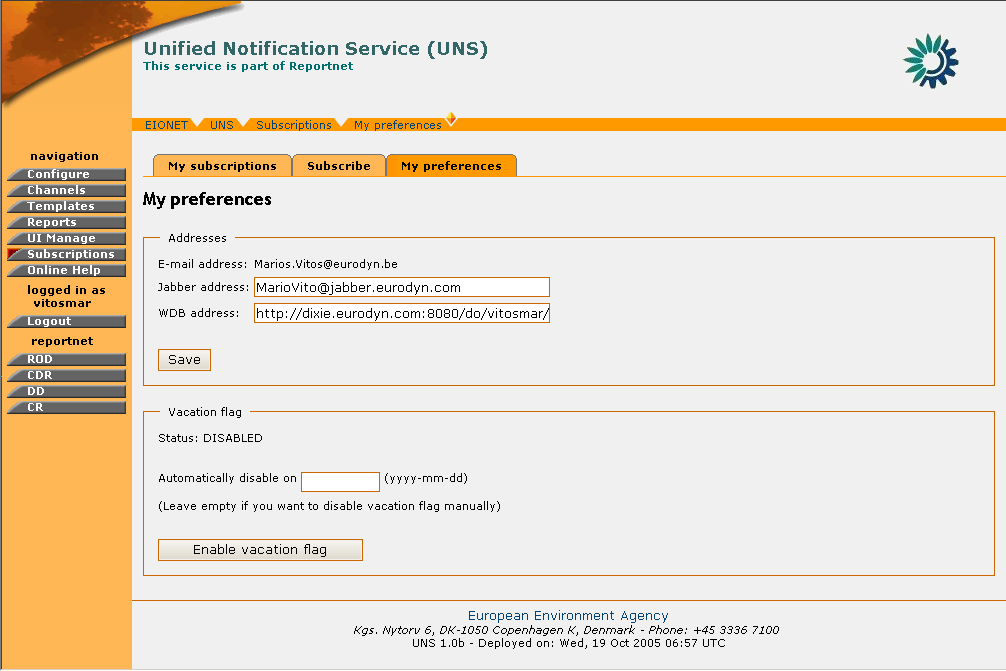
STEP 2 – Click on the “Subscriptions” button on the navigation bar.

By selecting the “Subscriptions” the user should select the “My Preferences” tab

STEP 3 – Enter/Edit a Jabber or WDB address for the delivery of the UNS notification messages.

**Important Note**: The user must also enter a valid Jabber account as well as a Web Dashboards address in order for the corresponding notifications to be delivered. Web Dashboards user address is the one that appears on the Internet browsers address bar when the user access to his/her dashboard at the Web Dashboards Web site. The e-mail address of the user cannot be changed and corresponds to the one defined in the EEA LDAP server. In case an invalid Jabber account or WDB address is inserted, the UNS will return the appropriate error message.

STEP 4 – Press “Save”

Figure 30: UNS - My Preferences page

So, in summary:

| Step | Setting Delivery Addresses |
| --- | --- |
| 1 | Login |
| 2 | Click on “Subscriptions” button |
| 3 | Enter Jabber/WDB addresses |
| 4 | Save |

#### Vacation flag

In cases that a user is away for a long period of time, he/she has the ability to enable a vacation flag notifying the UNS to stop sending notifications for a defined amount of time. In order to manage the vacation flag, the user needs to perform the following steps:

STEP 1 – Login

If you are not already logged in, see section 3 (Authenticate User Account).

STEP 2 – Click on “Subscriptions” button on the navigation bar.

By selecting the “Subscriptions” the user should select the “My Preferences” tab

STEP 3 – Set an automatic disable date (Optional)

Set a -future- date when the vacation flag will be automatically disabled. If the inserted date is not a future one, then the vacation flag won’t be enabled and the UNS will return the appropriate error message.

**Important Note**: If an automatic disable date is not provided, the vacation flag will stay enabled until it is disabled manually.

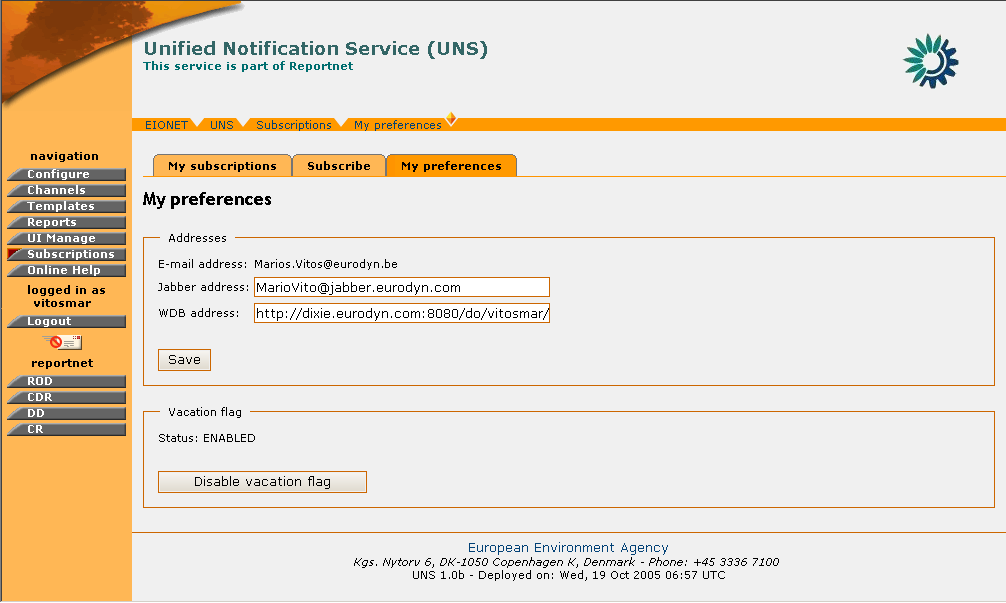
STEP 4 – Select the “Enable vacation flag” button.

Pressing this button will enable the vacation flag. The system will stop sending any notifications for harvested events to this user until the disable date of this setting. Subsequently, an image will be displayed for the UNS interface notifying the user that the vacation flag is enabled.

STEP 4 – Disable vacation flag

Press the “Disable vacation flag” button in order to start receiving notifications once again.   
The system will start delivering notifications generated from the moment of the disabling Vacation flag.

**Important Note**: Any notifications generated during vacation period will not be sent to the user after disabling Vacation flag.

Figure 31: Vacation flag enabled

### Subscribe to an Event channel

In order to subscribe to an event channel, you must be logged in UNS.

If you have the required user credentials, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3 (Authenticate User Account).

STEP 2 – Click on “Subscriptions” button on the navigation bar.

By selecting the “Subscriptions” the user should select the “Subscribe” tab

Figure 32: Event channels list

STEP 3 – Select channel to subscribe to

The “Subscribe” page displays all the channels that the user can subscribe to, according to his/her role. The available list of channels to the specific user will be related to the allowed roles as defined for each channel. The format of the specific notification event linked to the subscribed channel will be affected by the associated template which is also defined separately for each channel.

In order to perform a new subscription, the user must select the desired channel and then select the ‘Subscribe’ button.

STEP 4 – Enter Subscription Details

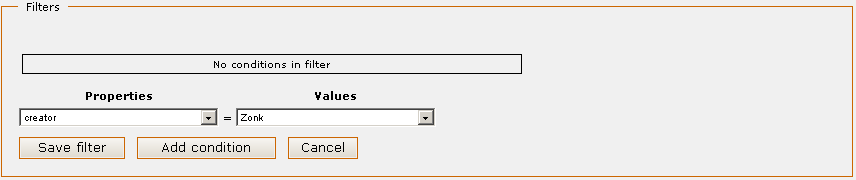
In order to complete the subscription, the user MAY specify how he/she wants to receive the notifications. The possibilities are directly related to the channel’s parameters. In other words the list of delivery types will be dynamically adjusted based on the channel’s settings.

E-mail, Jabber, and Web Dashboards (more than one selection is possible). The User must select some of the different delivery types in order to complete the subscription.

Figure 33: Subscription Details

STEP 5 – Define filters

In the Subscription page, the user also has the ability to filter the events for which he/she will receive a notification.

Figure 34: Filters menu

These filters enable the user to filter-out undesired data and receive notifications only for the event that matches the filter criteria.

In order to create a filter, the user must select the channel property according to which the filter will be based. The channel properties are identical to the channel Metadata (for more info on channel metadata see section 2).

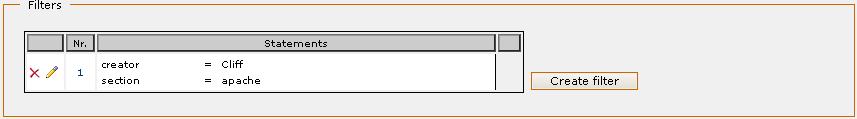
Then the user must select a target value for the filter. The list of values includes all the different values harvested for this specific property since the creation of the channel.

As soon as the property-value combination has been selected, the user may select “Add condition” and then define additional conditions for this filter. For subsequent conditions, a different property must be selected. The logical relation among different conditions of the same filter is “AND”. This means that for an event to pass from the filter, it should satisfy all the filter conditions.

When the filter conditions have been finalised, then the user must press the “Save filter” button in order for this filter to be created.

In the same manner the user may create multiple filters. In a case of multiple filters, an event that satisfies the criteria of any of the filters will be allowed to pass (“OR” logical relation).

The filters conditions can be changed / removed by pressing the “edit filter” button  on the desired filter. Likewise, for removing a filter, the user may select “delete filter” button  and remove it from the system.

Figure 35: Filter with multiple criteria

**Important Note**: The UNS does not allow two identical filters to be created. The UNS allows the creation of two overlapping filters but the user will be informed with appropriate warning message about overlapped filters.

STEP 6 – Complete Subscription

After all the required information has been inserted, the user must push the “Subscribe” button in order for the new subscription to be finalized.

**Important Note**: The user will receive notifications only for the events that will be harvested after the finalization of the subscription.

In summary,

| Step | Edit Notificator Configuration |
| --- | --- |
| 1 | Login |
| 2 | Click on “Subscriptions” button |
| 3 | Select a channel to Subscribe to |
| 4 | Enter Subscription Details |
| 5 | Define Exclude Filters |
| 6 | Complete Subscription |

### Edit a Subscription

In order to edit a subscription to an event channel, you must be logged in UNS.

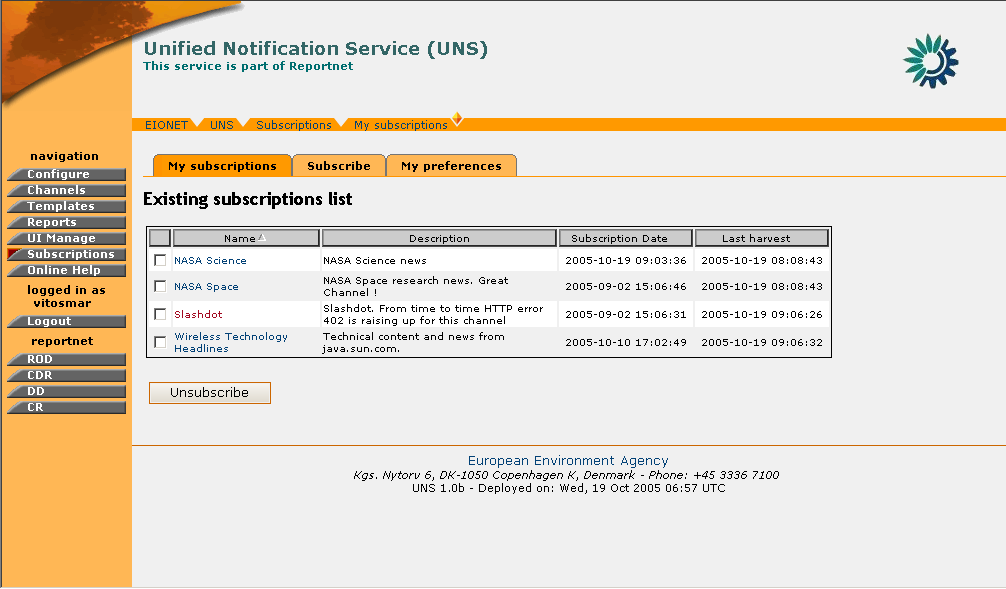
If you have the required user credentials, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

STEP 2 – Click on “Subscriptions” button on the navigation bar.

In order to perform an edit on a subscription, the user should select the “Existing Subscriptions” tab

Figure 36: Existing Subscriptions page

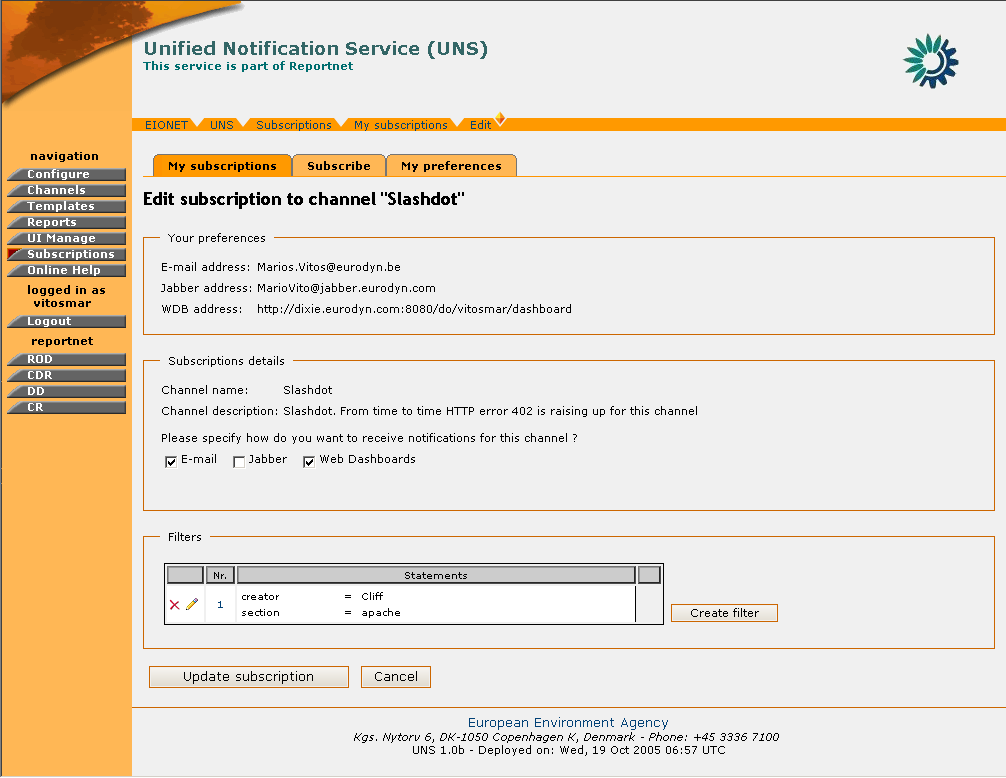
STEP 3 – Select a subscription to edit

The “My Subscriptions” page displays all the channels the user has subscribed to.

In order to edit an existing subscription, the user must select the desired channel by clicking on its name.

STEP 4 – Edit Subscription – Define filters

The user may edit the Jabber or WDB address as well as the desired notification delivery type(s).

Figure 37: Edit Subscription

In addition, the user may define additional filters or edit existing ones.

STEP 5 – Complete Subscription editing

After all the required information has been inserted, the user must push the “Update” button in order for the new settings to take effect.

In summary,

| Step | Edit Notificator Configuration |
| --- | --- |
| 1 | Login |
| 2 | Click on “Subscriptions” button |
| 3 | Select a Subscription to edit |
| 4 | Edit Subscription – Define Exclude filters |
| 5 | Complete Subscription Editing |

### Unsubscribe from an Event channel

In order to remove a subscription to an event channel, you must be logged in UNS.

If you have the required user credentials, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3 (Authenticate User Account).

STEP 2 – Click on “Subscriptions” button on the navigation bar.

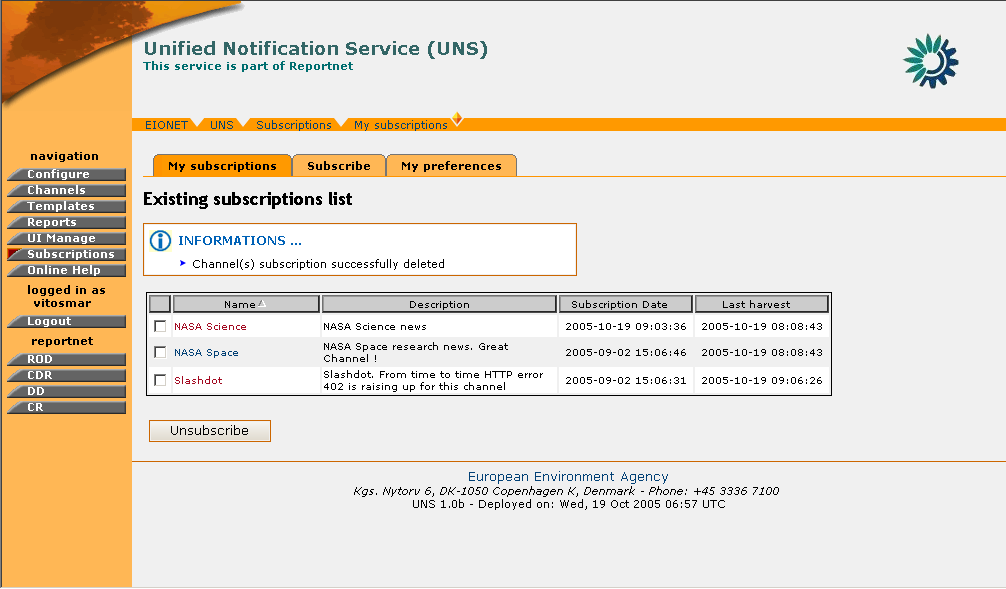
In order to unsubscribe from a channel, the user should select the “Existing Subscriptions” tab

STEP 3 – Unsubscribe from channel

The “My Subscription” page displays all the user’s existing subscriptions.

In order to unsubscribe from a channel, the user must select the desired channel and then select the ‘Unsubscribe’ button.

If the operation is successful, an information message will be displayed to the user

Figure 38: Information message after a successful delete of a subscription

In summary,

| Step | Edit Notificator Configuration |
| --- | --- |
| 1 | Login |
| 2 | Click on “Subscriptions” button |
| 3 | Unsubscribe from channel |

### Event Feeds

UNS allows the Reportnet users to view their events through an RSS feed. This content feed is available at: <http://dixie.eurodyn.com:9090/events> and Reportnet users can access this feed using any RSS/RDF reader software. The RSS feed contains only events from user selected subscriptions that use RSS feed as one of the delivery types.

# RPC user features

The group of RPC users represent external systems which interact with the UNS application.

These client applications may perform the following operations on UNS:

* create PUSH channels by using XML-RPC or Web UI,
* to push events to those channels by using XML-RPC and
* See list of their channels on the UNS web Site.

## UNS XML-RPC

In the context of the UNS, XML-RPC can be used to send RDF formed events data to the UNS, in order to create PUSH channels that will be used for the insertion of events as well as to subscribe users. Those operations are allowed only to the registered UNS RPC users. UNS RPC users are allowed to operate only on channels created by them.

The UNS XML RPC Router is: http://<<web\_root>>/rpcrouter

There are currently five procedures available to the UNS RPC users:

1. **UNSService.sendNotification(channel\_id, triples)**  
   Send events data to the specified UNS's PUSH channel.

*PARAMETERS* **:**

*channel\_id*string - represents channel identifier inside the UNS.

*triples* array - events data as straightforward *triples* in a list-of-lists structure

1. **UNSService.sendNotificationRDF(channel\_id, rdf)**  
   Same as method *sendNotification*. Difference is the *rdf* parameter. In this case that parameter is a string representing RDF.
2. **UNSService.createChannel(channel\_name, description)**   
    Creates new PUSH channel. It takes as parameters the name of the new  
   channel (required) and its description (the phrase describing the purpose of the channel)

*PARAMETERS* **:**

*channel\_name* ***s***tring – channel name

*description* string – terse phrase describing the channel

*RETURNS* *:*

*string* String representing new channel's identifier

1. **UNSService.canSubscribe(channel\_id, username)**Checks if specified user is allowed to make subscription to the specified channel  
   PARAMETERS**:**

*channel\_id*string - represents channel identifier inside the UNS.

*username* string - EEA's LDAP (Synchronized with CIRCA) username

*RETURNS*:

*boolean* indicating if user is allowed to make subscription to the specified channel.

True - if user is allowed

False – otherwise

1. **UNSService.makeSubscribtion(channel\_id, username, filters)**Creates new or updates existing subscription to the specified channel for the specified user. If subscription for provided channel exists already it will be updated, otherwise it will be created.

*PARAMETERS* **:**

*channel\_id*string - represents channel identifier inside the UNS.

*username* string - EEA's LDAP (Synchronized with CIRCA) username

*filters*array - An array of the dictionaries in form predicate-object representing user’s filters   
 *Filter* *Example*:

filter = [{

'http://rod.eionet.eu.int/schema.rdf#locality': 'DK',

'http://rod.eionet.eu.int/schema.rdf#issue': 'Water'

'http://rod.eionet.eu.int/schema.rdf#obligation': 'Some obligation'

}]

In case of errors UNS will return common HTTP status codes:

* “ServiceUnavailable" 503 (Will be returned in case that remote service is trying to push data to the existing but disabled channel)
* "NotFound" 404 (Will be returned in case that remote service is trying to push data to the non existing channel or if tries to subscribe a non existing user)
* "BadRequest" 400 (Will be returned in case that remote service is trying to provide bad input to the UNS)
* "InternalError" 500 (Will be returned in case of internal UNS errors)
* "Unauthorized" - 401 (Will be returned if remote service is unauthorized to perform desired operation)
* “Method Not Allowed” (Will be returned if the remote service tries to operate on a channel that does not belong to it)

To make use of the UNS XML RPC, users need to call it from an application. The following is an example for such an RPC call using the Python programming language:

import xmlrpclib

server = xmlrpclib.Server(‘http://<<web\_root>>/rpcrouter’,

BasicAuthTransport(user,password))

try:

server.UNSServer.sendNotificationRDF(‘cdr’,rdfMessage)

if server.UNSServer.canSubscribe(‘cdr’,’someuser’):

server.UNSServer.makeSubscription(‘cdr’,’someuser’,[])

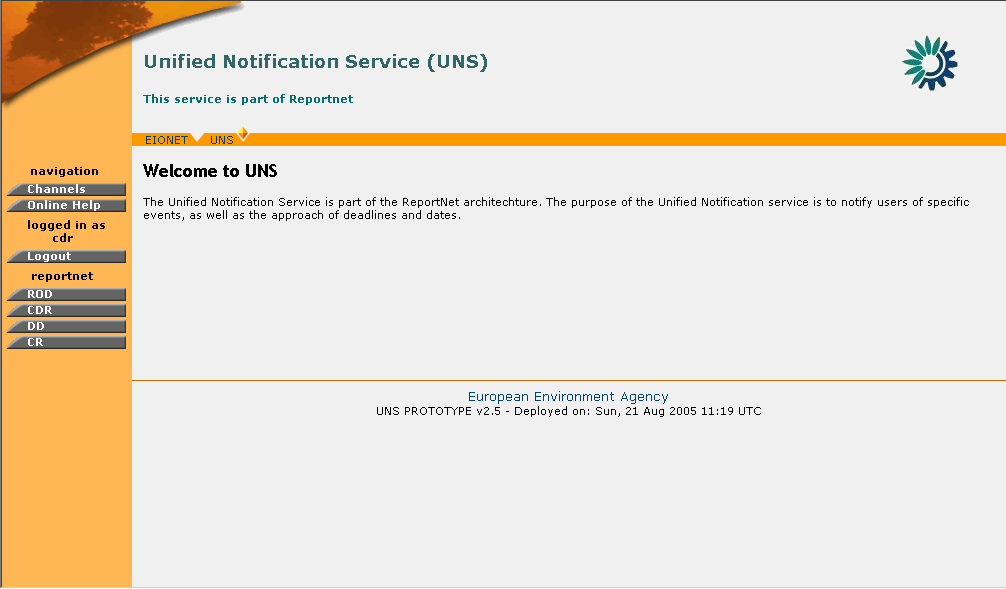
except xmlrpclib.ProtocolError, p:

err\_code=p.errcode

#handle error according to error code

## UNS RPC Users UI

As mentioned in the introduction of this section, the RPC users may view only the list of the channels they have created, or create a new PUSH channel.

Figure 39: RPC User welcome screen

### RPC User PUSH channel Creation

In order to create a PUSH channel, you must be logged in UNS as RPC User.

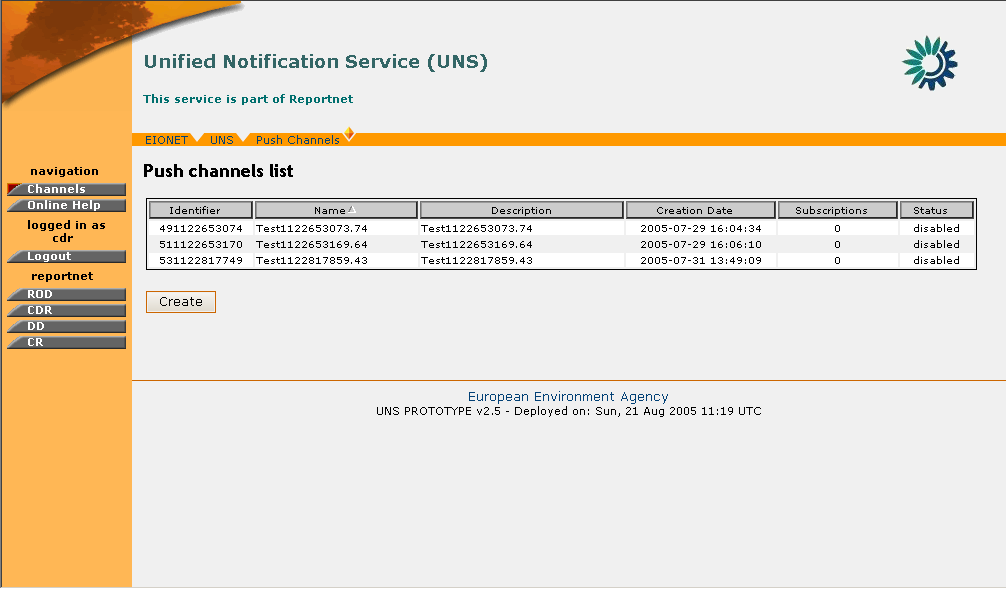
If you have the required user credentials, then the following steps must be followed:

STEP 1 – Login

If you are not already logged in, see section 3.

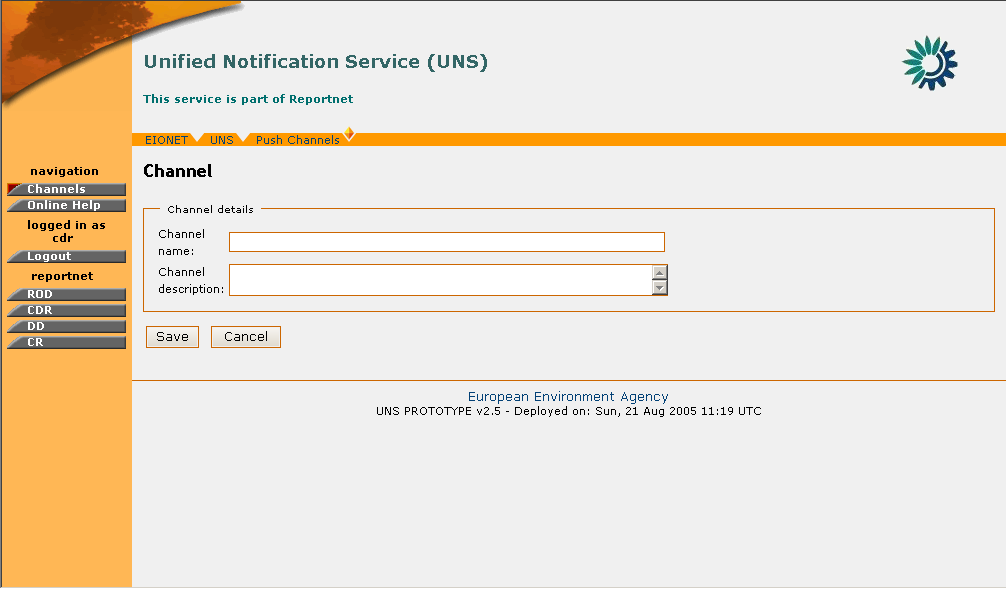
STEP 2 – Click on “Channels” button on the navigation bar.

This will take you to the list of the existing PUSH channels for the corresponding user. The identifier of the channel which is required for insertion of events as well as the channel status are the two additional columns displayed in this list.

Figure 40: PUSH channels List

STEP 3 – Click “Create”

By clicking on the Create button, the user arrives at the PUSH channel creation dialogue. A channel name and description will be required to introduced by the RPC user in order for the channel to be created.

Figure 41: PUSH channel Creation form

STEP 4 – Finalize PUSH channel creation

After all the required data have been inserted, the user must press the “Save” button in order for the channel to be created.

So, in summary:

|  |  |
| --- | --- |
| Step | RPC User PUSH channel creation |
| 1 | Login |
| 2 | Click on “Channels” button |
| 3 | Click “Create” button |
| 4 | Finalize PUSH channel creation |

1. This list is not exhaustive. The full RSS specification can be found at <http://blogs.law.harvard.edu/tech/rss> [↑](#footnote-ref-2)
2. Complete specifications for RDF are found at: <http://www.w3.org/RDF/> [↑](#footnote-ref-3)
3. Dynamic content is considered to be the channel content that changes from event to event (date, title, message body etc.) [↑](#footnote-ref-4)